ARMY RESEARCH LABORATORY

Hyperspectral and Spectro-Polarimetric Pixel-Level Classification Using Genetic Programming

Patrick Rauss

ARL-TR-907

November 2001

Approved for public release; distribution unlimited.

Army Research Laboratory

Adelphi, MD 20783-1197

ARL-TR-907 **November 2001**

Hyperspectral and Spectro-Polarimetric Pixel-Level Classification Using Genetic Programming

Patrick Rauss

Sensors and Electron Devices Directorate

Abstract

The objective force will be relying heavily on their sensors to be a combat multiplier to help improve the force's effectiveness and survivability, particularly for reconnaissance, surveillance, and target acquisition missions. Currently, fielded passive sensor systems are generally ineffective against camouflage, concealment, and deception. Their performance is also sensitive to environmental conditions. To meet future needs, several new sensor systems are being developed and evaluated. Two of these new sensors are passive systems that collect additional, measurable characteristics of light: hyperspectral (HS) systems and spectro-polarimetric (SP) systems.

To fully take advantage of the information that these systems collect requires new algorithms and techniques. This report discusses why new techniques are necessary and details the development of a computer-assisted design system for the discovery of classification algorithms via a small number of sample target and background signatures. The technique is called genetic programming (GP). GP is an adaptive learning technique that automatically generates a computer program (in this work, a mathematical equation) to solve the problem it is given.

This report documents work conducted primarily between September 1999 and August 2000, while the author was on a rotation at the University of Michigan under the Federated Laboratories Consortium program. The report demonstrates that GP could be a useful technique for processing HS and SP data. The experiments reported here show that by using even the simplest of operators (addition, subtraction, multiplication and division) the GP process can develop interesting and potentially useful solution equations. The results shown here are encouraging. However, many questions remain to be answered.

Contents

Executive Summary 1

1. Introduction 3

2. Background 5

2.1 The Sensors 5

2.2 The Software 10

2.2.1 Broadband Analytical Techniques 11

2.2.2 Multi-band Analytical Techniques 12

2.2.3 Hyperspectral Techniques 14

2.2.4 Spectro-polarimetric Techniques 17

3. Genetic Programming 18

3.1 GP in Image Processing 21

4. GP for Hyperspectral and Spectro-Polarimetric Data 23

4.1 System Design 23

4.2 Imagery Used 24

4.3 Spectral Image Cube 24

4.4 Target Classes 25

4.5 Training Pixel Extraction 26

4.6 GP Software and Parameter Selection 29

4.6.1 The GP Software 29

4.6.2 Parameter Selection 29

4.6.3 Batch Runs 31

4.7 GP Output 32

4.8 The Parser 33

4.9 Evaluate Batch Results 33

4.9.1 Protected Division in MATLAB® 35

4.10 Processing Full Images 35

5. System Testing 36

5.1 Lil-gp Multiple Variable Evaluation 36

5.2 System Evaluation 36

5.3 Real Data: 28 Bands 37

5.4 112 Bands 38

5.4.1 Fitness Function: Thresholding; Target > 0, Non-Target < 0 39

5.4.2 Fitness Function: Thresholding; Target > 1, Non-Target < –1 40

5.4.3 Fitness Function: Bracketed; 1<Target<5, –5<Non-Target<–1 40

5.4.4 Fitness Function Evaluation 40

5.5 Addition of Constants as Terminals 40

6. Evaluation Image Results **43**

6.1 Processing the Entire Training Image 43

6.2 Processing Evaluation Images 44

7. Conclusions **47**

7.1 Areas for Future Investigation 47

Acknowledgments **48**

References **49**

Distribution List **69**

Report Documentation Page **71**

Appendices

A. Best Equations **53**

B. Processed Images **57**

Figures

1. LandSat TM images of Moro Bay, CA January 25, 1988 7
2. False color composite of Moro Bay and true color composite 7
3. Broadband image of HS scene with six targets 9
4. Processed HS image clearly revealing four of the six targets 9

Figures (cont'd)

5. Why HS systems are more sensitive to contamination and noise effects 15

6. Example of mutating an operator and a terminal 19

7. Mutation of tree through replacement of one branch with a copy of another 19

8. Cross-over operation 20

9. Data flow of current system 23

10. Band 72 of AOTF image from which the training pixels were extracted 25

11. Single band of AOTF spectral image showing approximate areas where training pixels for grass and pad classes were extracted 26

12. a and b: Enlargements of Wall and Barrel with approximate location of training pixels marked 26

13. Raw intensity values of 20 pixels from each class 27

14. Normalized training spectra for each class 28

15. Example of symmetric and asymmetric trees 30

16. Example of a simple .bst file from a lil-gp run 32

17. Image chips used to evaluate batch generated equation performance 34

18. User defined mrdivide.m function for doing protected division in MATLAB® 35

19. System evaluation image 36

20. Best-of-run equation using values in figure 19 as input 38

21. Parser results for the previous equation 38

22. Best-of-run results on single polarization image 39

23. Parsed GP equation for single polarization AOTF image 39

24. Classification image for grass using equation from single polarization GP run 39

25. Classification images for barrel bright; GP equation 5 43

26. Parsed GP equation 5 used to generate figure 25 43

27. Classification images for Grass; GP equation 1 44

28. Parsed GP equation used to generate figure 27 44

29. Evaluation image *Fence* 45

30. Evaluation image *Trucks* 45

31. Evaluation image *Van* 46

Figures (cont'd)

32. Evaluation image Trucks processed with *barrel-dark* equations 46

33. Evaluation image Trucks processed with *barrel-bright* equations ... 46

Tables

1. LandSat TM bands 6

2. Several important HS systems 8

3. Differences between MS and HS imaging systems 10

4. Demonstration of the rapid growth in the number of images possible for HS and SP systems 13

5. Data file used for GP software development and evaluation 36

6. Training values from evaluation image presented to GP to validate modifications for multiple variables 37

7. Number of equations that used the new terminals 41

8. Number of "useful" equations that used the new terminals 42

Executive Summary

Passive sensor systems will be key tools of the lighter, faster Army. However, currently fielded passive systems are highly susceptible to camouflage, concealment, and deception and environmental conditions—conditions likely to be encountered by this force. Further, the Army has recognized that existing systems are inadequate. The lighter forces will be relying heavily on their sensors to be a combat multiplier, particularly for reconnaissance, surveillance, and target acquisition and indications and early warnings missions, to help improve the forces effectiveness and survivability. To that end, several new sensor systems are being developed and evaluated to assist in these missions. Two of the sensors collect additional, measurable characteristics of light: hyperspectral (HS) systems and spectro-polarimetric (SP) systems. HS systems record not only the intensity of light but also its wavelength, thus allowing the detection of a spectral signature of the materials in the scene. SP systems record not only the spectrum of the materials but also the polarization of the light detected. These new systems produce orders of magnitude more data per frame than traditional systems. The data must be processed into a useful form to be useful to the soldier. Current techniques for processing HS and SP data are either too complicated, requiring expert user interaction, or inaccurate (susceptible to false alarms) to be useful in the field.

To fully take advantage of the information these systems collect requires new algorithms and techniques. This report discusses why new techniques are necessary and then details the development of a computer-assisted design system for the discovery of classification algorithms via a small number of example target and background signatures. The technique is called genetic programming (GP).

GP is an adaptive learning technique that automatically generates a computer program (in this work, a mathematical equation) to solve the problem it is given. GP is a technique to stochastically search the entire space of possible solutions to find a solution to the problem presented. Being a branch of genetic algorithm research, GP develops a solution by using operations similar to biological evolution: reproduction of an individual solution, crossover between two individuals, and mutation of an individual solution. Starting with an initial population of potential solutions, GP uses these evolutionary operators to build better solutions.

This report demonstrates that GP could be a useful technique for processing HS and SP data. The experiments reported here show that with the use of only the simplest of operators (addition, subtraction, multiplication, and division), interesting and useful solution equations can evolve through the GP process. Further, only very simple fitness functions were used to generate these results. Based on these two results, with extension of the types of operators used and the inclusion of numerical constant values, better performing equations can be expected. It is clear that further research is needed into the definition of the fitness function. One goal of investigating the fitness function would be to develop a method to generate a confidence value for the classification results. The results shown here are encouraging. However, many questions remain to be answered.

One difficulty encountered that will have significant impact on future efforts is the apparent lack of adequate ground-truthed development and evaluation data to allow training and quantitative evaluation of a techniques performance. Until such data become available, it will be difficult to quantitatively measure the performance of any technique or to compare the performance of different techniques.

1. Introduction

An important factor for improving the soldier's survivability in the field is the speed, range, and accuracy with which he is able to find and identify targets: friendly, hostile, and in today's environment, neutral. One method to assist the soldier in this task is to use "sensors" to extend his capabilities. Commonly, these sensors only collect information and present it to the soldier as an image. The soldier still must evaluate the image to find and identify the targets in it. The speed at which the soldier can accurately process the images covering a fixed field of view (FOV) appears to be decreasing. This is probably attributable to several factors:

- The increasing number of sensors available,
- The increasing number of images that a single sensor presents to the soldier (frames per second), and
- The increasing level of information available in each of the individual image frames presented.

This decline in performance is clearly unacceptable in tactical situations where seconds may determine survival.

Clearly, automated processing must be employed to assist the soldier in processing the large quantity of information available. Automatic target recognition (ATR) systems have the potential to increase the speed at which the soldier can process this imagery. Whether an automated system simply cues the soldier or provides a location and target identification, speed and accuracy are key features that such systems need to possess. Tactically, a system that is 100 percent accurate but takes an hour to process each frame is no more valuable than a system that runs in real time but is only 50 percent accurate.

A number of sensors are available on the battlefield today. The most common types are passive systems. These passive systems record only the energy emitted by the scene and/or energy from the environment (commonly, the sun) reflected from the scene. Two common examples are the forward looking infrared (FLIR) camera, which records the thermal energy emitted by the scene and television (TV) which records visible light reflected from the scene. This information is presented to the soldier as an intensity image. The more energy recorded for a pixel, the more brightly the pixel is displayed.

Automatic processing of images from these sensors relies on the extraction of spatial features of some sort from the imagery. A spatial feature is a local collection of pixels that stand out in some noticeable way from the other pixels around them. These pixels stand out because they are contrasted with their neighbors. Unfortunately, situations often occur when there is not enough contrast in the image to find these spatial features.

Low contrast situations are likely to occur with even greater frequency in environments into which the Full Spectrum Brigade is likely to be deployed. The brigade will rely heavily on its reconnaissance, surveillance, and target acquisition (RSTA) squadron for situational awareness and indications and early warning (IEW) activities. The Army is well aware of current sensor suites' shortfalls and recognizes that a more robust sensor suite is needed to enhance the RSTA squadron's capabilities.

Other passive sensors can collect additional information beyond the light's intensity. One type of sensor collects spectral information, that is, the intensity of the light over small wavelength ranges. Using spectral information, one can discriminate between materials that are indistinguishable in a broadband image, such as FLIR or TV. Additionally, if the spectral resolution is high enough, materials can even be identified. Another characteristic of light that may be useful to exploit for discrimination and detection purposes is the light's polarization.

To develop and use new hyperspectral (HS) and spectro-polarimetric (SP) applications requires some level of automated processing. A challenge now is to develop automated analytical techniques to take advantage of HS and SP systems, to assist users in using this new capability, and to process the vast amounts of data that these sensors will make available. In addition, since HS and SP applications are relatively new, it would benefit the community to have available development tools to aid in determining useful spectral features and/or polarizations for particular tasks.

This report documents work done in applying an adaptive learning technique called genetic programming (GP) to HS and SP imagery. This technical report shows that GP has the potential to be a key tool in image processing and feature extraction. The goal of this research project was to construct a computer-assisted design system for the discovery of classification algorithms and germane band information via a small number of example sensor data from user-defined classes.

The next section explains why new techniques are necessary by reviewing some background about the sensors and techniques currently available. Section 3 defines genetic programming. Section 4 details the system designed for this effort. Section 5 documents evaluation and experimented in-scene results. Section 6 addresses results with three evaluation images. Finally, section 7 discusses the project results and outlines where to go from here.

2. Background

2.1 The Sensors

A number of sensors are available on the battlefield today. The most common types are passive systems. These passive systems record only the energy emitted by the scene and/or the energy from the environment reflected from the scene. As mentioned earlier, two common examples are FLIR, which records the thermal energy emitted by the scene, and TV, which records visible light reflected from the scene.

Both FLIR and black-and-white TV (BWTV) are considered broadband systems. They record light over a broad range of wavelengths. FLIR generally is sensitive to the thermal infrared range of 8 to 12 microns while BWTV covers the visible range from ~0.4 to 0.8 micron. ATR algorithms designed to process broadband data rely on the contrast between objects for detection. Most ATR systems use spatial features extracted from these contrast areas for detection and recognition (e.g., Der and Chellappa, 1997; McKee and Bandera, 1998; Wang, Der, and Nasrabadi, 1998; and Hecht-Nielsen and Zhou, 1995). If there is no contrast, these spatial features cannot be extracted. For example, with FLIR systems, objects are difficult to resolve as separate entities when both have similar apparent temperatures.

This condition occurs more often than one might think. For example, assume that the object to be detected is a vehicle. If the vehicle is not active, it will reach a thermal equilibrium with its environment. Differences in the thermal mass and the thermal conductance of targets and background objects allow natural outside energy sources, such as the sun, to affect the surface temperature of the object. These differences in thermal mass and conductance are reflected in the different rates of change in each object's apparent temperature. Objects such as leaves and grass have low thermal mass and high conductance and thus change temperature rapidly to reflect current energy flux conditions. On the other hand, a large metal vehicle has a large thermal mass and a high thermal conductance and it therefore heats and cools slowly; yet its surface responds rapidly to direct sunlight by warming rapidly. This leads to conditions where near-zero contrast events occur shortly after dawn and dusk as the vegetation warms or cools rapidly and the vehicle does not. For a period of time, the vegetation and vehicle have the same apparent temperature. For a short time during these "cross-over" times, there is zero contrast. Therefore, no spatial features can be extracted. Beyond this and other natural contrast reductions, spatial features can be and are intentionally obscured through the use of camouflage, concealment, and deception (CC&D) by either obscuring the feature itself or reducing the contrast so that the feature cannot be detected in FLIR or BWTV imagery.

One method for discriminating between objects that look similar in broadband imagery is to use multi-band or multi-spectral (MS) imagery. For example, color television collects images in three narrower bands (blue, green, and red), while seven spectral bands are collected with the LandSat thematic mapper (TM), six bands covering regions from the blue to the near IR and one in the thermal IR (see table 1). By comparing intensities in different bands, we can separate objects or materials that are indistinguishable in a broadband image. Figure 1 shows images collected by the LandSat TM sensor. The four images show Bands 1 through 4. Close examination reveals some differences between the images, particularly in the water areas near the beaches. However, it is very difficult to easily discriminate different land use areas in the individual band images.

Figure 2 shows two different color images created with three of the images from figure 1. Figure 2(b) is a "true color" image, which renders a color photograph of the scene. Figure 2(a) is a "false color" representation where the RGB bands of the image represent the next higher spectral band of the TM sensor. Blue in the image represents the green band of the sensor. Green in the image represents the red band of the sensor and red in the image represents the near IR band in the image.

Historically, this false color band mapping to the RGB pallet is used to reproduce the pallet obtained from IR photographic film. Its importance is that it highlights vegetation in shades of red. This is an example of using a broad spectral feature or a spectral signature to discriminate materials. This spectral approach can be used to find materials of interest and highlight them and suppress materials such as grass or trees. This improves the contrast of objects of interest. These enhanced images can then be spatially processed in a more traditional manner.

MS imagers for remote sensing have been in operation since at least the mid-1970s. Since that time, they have proved their value in many applications, such as crop yield estimation, forest harvest monitoring, evaluation of soil conditions, crop and timber type identification, regional planning, and cartographic revisions.[1] However, since MS systems have inherently

Table 1. LandSat TM bands.

Band no.	Wavelength interval (µm)	Spectral response	Spatial resolution (m)
1	0.45–0.52	Blue-Green	30
2	0.52–0.60	Green	30
3	0.63–0.69	Red	30
4	0.76–0.90	Near IR	30
5	1.55–1.75	Mid-IR	30
6	10.40–12.50	Thermal IR	120
7	2.08–2.35	Mid-IR	30

[1] For an extensive table of LandSat 7 applications, see http://landsat. gsfc.nasa.gov/ images/ Landsat_Applications.html

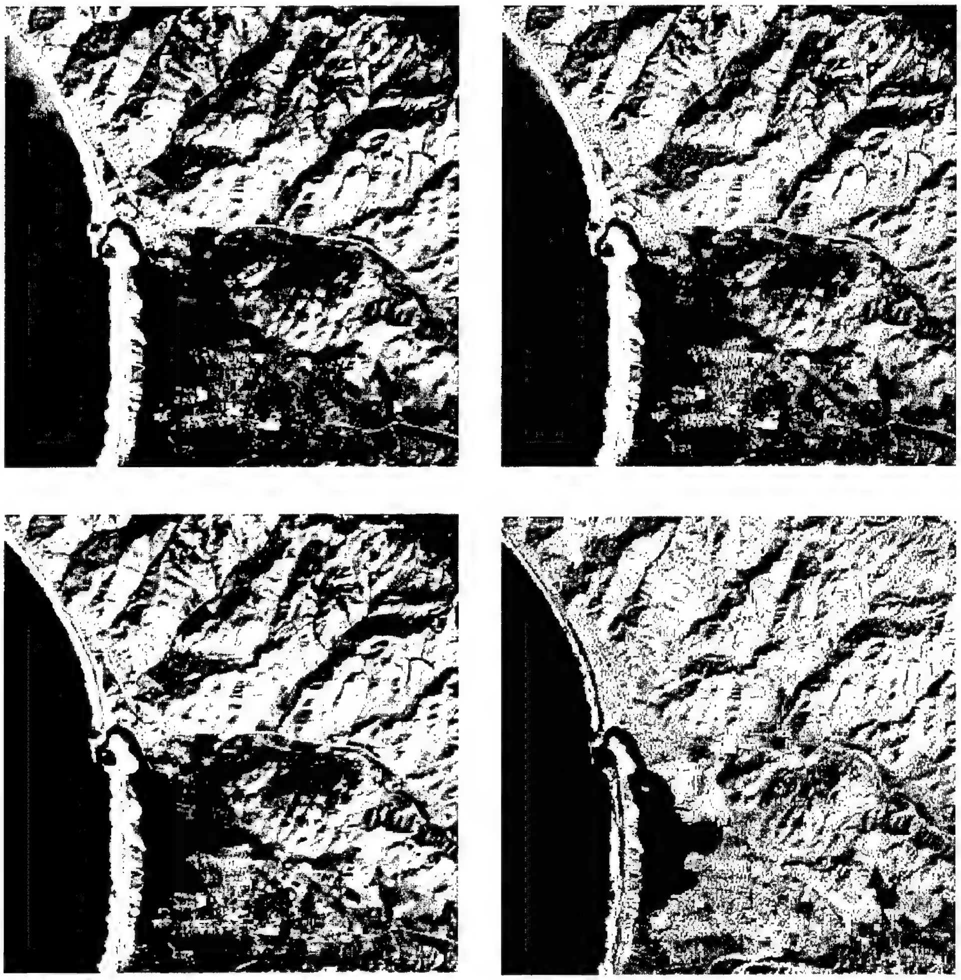


Figure 1. LandSat TM images of Moro Bay, CA January 25, 1988. (Top left band 1, top right band 2, bottom left band 3, and bottom right band 4.)[1]

[1] Images in figure 1 taken from http://rst.gsfc.nasa.gov/Sect1/Sect1_3.html

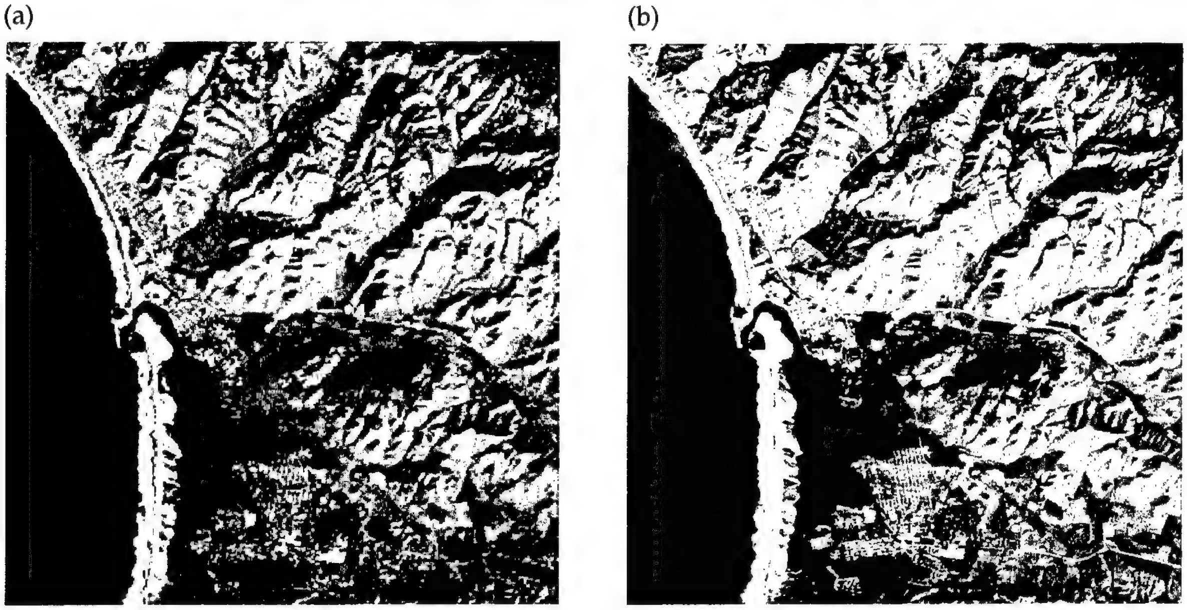


Figure 2. (a) False Color composite of Moro Bay (Red = band 4, Green = band 3, Blue = band 2.) (b) True color composite image (Red = band 3, Green = band 2 and Blue = band 1.)[2]

[2] Images in figure 2 taken from http://rst.gsfc.nasa.gov/Sect1/Sect1_8.html and Sect1_10.html

broad spectral resolution, their usefulness is limited to discriminating between objects with significantly different and broad spectral characteristics, such as trees from dirt or red flags from green flags. Objects that differ only at finer spectral resolutions cannot be distinguished easily, if at all, such as oak trees from pine trees or pink flags from red flags. Sensors of these types would be of limited value in tactical situations. However, they do suggest an approach to suppressing clutter and detecting targets. The suggestion is that the use of more and finer spectral bands may allow discrimination of spectrally similar materials.

Typically, a system with a bandwidth on the order of 20 nm or less and with 100 or more bands of contiguous coverage over the system's spectral range is considered to be an HS system, although other definitions are commonly found.

Although current HS imagers are primarily experimental research systems, they are becoming more common. A number of airborne systems exist and several satellite systems are scheduled for launch in the near future.[2,3] Table 2 shows a few HS systems currently available. For a more extensive list of HS systems, see appendix D of the Remote Sensing Tutorial at http://rst.gsfc.nasa.gov.

With spectral resolution of this level, it is possible to identify materials in the image by their spectra. Using these spectral signatures, we can identify and suppress natural materials in a scene, allowing spatial detection of objects of interest. Further, identification of materials such as metal, paints, textiles, and so forth can be used to highlight areas or objects of interest in a scene. IR HS systems can also be used for the identification of many chemicals.

Another characteristic of light that can also be exploited for discrimination tasks during certain circumstances is the polarization of the light (Guenther, 1990; Slater, 1980; Wolff, 1995). The reflection (in the visible) and emission

Table 2. Several important HS systems.

System	Sponsor	Year fielded	Spectral range microns	Number of bands	Approximate bandwidth of each band (nm)
AVIRIS	NASA	1987	.4 to 2.5	228	10
HYDICE	NRL	1995	.4 to 2.5	206	10
SEBASS	AFRL	1997	7.4 to 14	128	10
FTHSI	AFRL	2000	.35 to 1	256	1.7

[2] The Moderate Resolution Imaging Spectroradiometer Proto-flight model was launched in December 1999. First-light data were collected in February 2000. With 36 bands of bandwidths from 20 to 300 nm., it is considered by some an HS system.

[3] AFRLs MightSat II.1 was launched in July 2000 with an imaging Fourier transform spectrometer aboard.

(in the thermal) characteristics of material surfaces can define the light's polarization. This polarization may prove useful for discrimination tasks by affecting the contrast between materials. Additionally, environmental conditions (moisture, multiple reflections, multi-path reflections, etc) can alter the light's polarization, which may prove useful for determining some atmospheric conditions of the transmission path.

Recently, some experimental imaging sensors have been constructed, which collect spectral information as well as polarization information for each spectral band. These imaging SPs or HS polarimeters have the potential to assist in discriminating different materials in an image through both spectral and polarimetric means.

Using the additional information supplied by HS and SP imagers, we can detect targets even when there is little or no broadband contrast with the local background, we can suppress false alarms, and we can highlight and identify CC&D (Rauss, Cederquist, Dwan, and Wegrzyn, 1999). These tasks can all be performed basically in the same way, by identifying the material(s) in each image pixel via its spectral or SP signature. Once processed in this manner, the image can be reconstructed to highlight materials of interest and to suppress those not of interest (Rauss, Cederquist, Dwan, and Wegrzyn, 1999; Rauss, Daida, and Chaudhary, 2000). This identification can be used to generate a classification image with better contrast between objects. These new image maps can then be spatially processed in more common ways. Figures 3 and 4 show an example of the advantage that an HS sensor can give with the right processing. Figure 3 shows a simulated FLIR image that was created by the summing of all the bands from an IR HS

Figure 3. Broadband image of HS scene with six targets.

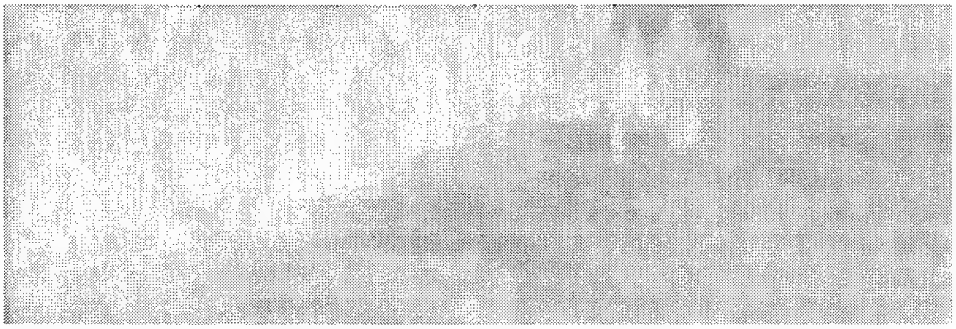
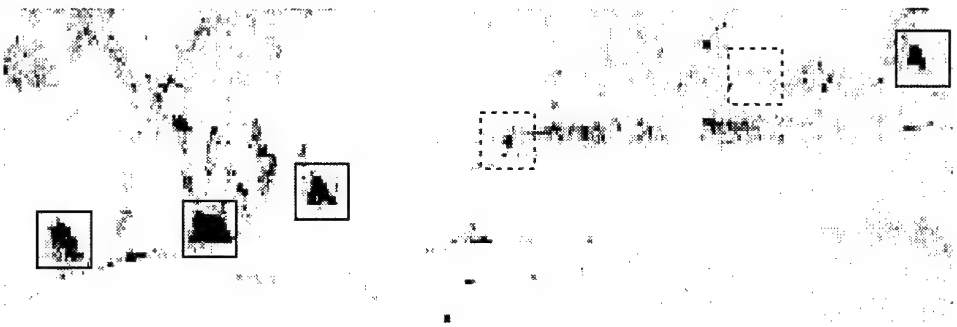


Figure 4. Processed HS image clearly revealing four of the six targets. (Boxes added to show target locations. Dashed boxes indicate missed targets.)



imager. There are six targets in the image. Obviously, the contrast level is very low. The targets are in a high clutter environment and CC&D is in use.

Figure 4 shows the results of performing advanced HS processing on the spectral image cube. Four of the six targets are now clearly visible. Much of the clutter has been suppressed, and this image could be spatially processed to find at least the four obvious targets and suppress much of the remaining clutter. Figure 4 also points out that HS sensors will not necessarily allow recognition of the target. In this case, all the targets are under camouflage nets and only the silhouette of the nets can be seen. Identification of which target (if any) is under the nets is not possible.

Experimental HS and SP sensors are beginning to collect research and development data in the visible and the IR ranges. One difficulty is the tremendous increase in the amount of information collected for each pixel with such systems. Table 3 shows some of the typical differences between MS and HS systems; particularly note the size of the standard data files.

The hardware is proving to be viable but to be truly useful, the data must be processed into a user-friendly form, whether the user is the next algorithm in the system or a soldier in the field.

2.2 The Software

Data analysis techniques lag behind hardware development for several reasons. First, only in recent years have computers become commonly available that are capable of processing the huge amounts of data collected by HS and SP systems in useful time frames. Second, the information content of the sensors has gone from simple, in the broadband case, to highly complex in an HS or SP image. ATR algorithms for broadband imagers such as FLIR were developed with only the relatively simple information content available. While the application of these techniques to MS, HS, or SP systems may show some evolutionary improvement in performance, significant improvements are unlikely without taking advantage of the additional information content found in these data streams. Finally, to be effective,

Table 3. Differences between MS and HS imaging systems. (AOTF refers to the system whose data are used in this report, see sect. 4.2 for details.)

System features	Multispectral	Hyperspectral
Number of bands	7 (LandSat TM) Thematic mapper	224 (AVIRIS) 112 (AOTF)
Spectral resolution (nm)	450 to 1250 (LandSat)	10 (AVIRIS) 10–20 (AOTF)
First deployment	1972 (LandSat 1)	1987 (AVIRIS) 1997 (AOTF)
Common deployment	Satellite	Airborne (AVIRIS) Ground (AOTF)
Typical single image cube file size (MB)	2.1 (LandSat TM)	140 (AVIRIS) 34 (AOTF)

algorithm development needs data from actual sensors. Therefore, it is necessary to wait until the sensors progress beyond the early experimental stage and begin collecting realistic data before experimental algorithm development and testing can take place. Furthermore, to develop and evaluate algorithms for realistic conditions, algorithm developers must wait for sufficient quantities of accurate ground truthed data to be collected with the actual hardware. The data must cover realistic variations in content and quality to allow proper algorithm evaluation. Data collected in a hot desert environment will probably not adequately support development of an algorithm to be used in tropical or temperate environments.

The potential value of HS and SP imaging is high, but the number of existing applications is presently small for several reasons. First, the imagers are relatively new, so applications unique to them have yet to be fully identified and developed. Second, many of the existing applications are based on MS approaches. These adaptations often do not exploit the strength of HS data, or their fine spectral resolution, and do not even address polarization. This results in only incremental performance improvements over the MS applications. Third, tools to investigate new applications for HS and SP imagery are not common. Fourth, most existing tools for exploiting HS imagery require extensive knowledge of spectroscopic techniques. This expertise is uncommon for many of the potential users and application developers. Finally, the data files themselves can be very intimidating, easily reaching gigabytes in size for even small data collections.

2.2.1 Broadband Analytical Techniques

An extensive body of work involving detection and recognition of objects in single band or color (three-band) imagery exists. Military research in the field is limited, when compared to the amount of work done in computer vision and commercial image-processing techniques. Many books, journals, and newsletters are dedicated to computer and machine vision, image processing, image understanding, and so on.

However, the quality of the military research is at the high end of the range of research performed or conducted. This is primarily attributable to the difficulty of the military problem. Performance levels must be extremely high, as already mentioned, and many of the more common approaches cannot perform at those levels. This leads military research to use the most cutting edge techniques such as neural networks, Markov fields, wavelets, image understanding, and other highly sophisticated approaches.

Most of this military work has been conducted primarily with radar in its various forms, FLIR imagery, and to a lesser extent, laser radar. The Society of Photo-Optical Instrumentation Engineers AreoSense conference regularly has multiple sessions dedicated to target detection and target recognition research and development. An ATR Working Group, a joint Government (primarily DoD), and industrial (primarily defense contractors) group also discusses and presents classified work in target detection and recognition R&D.

With the research that has been applied to passive ATR efforts, it is becoming more and more difficult to achieve higher levels of performance with these methods. During certain conditions, these techniques can achieve useful levels of performance. Unfortunately, the conditions necessary for adequate performance levels are somewhat limited. The community has been focusing in recent years on augmenting single broadband sensor techniques with other sensors in a data fusion approach, which can be thought of as a MS approach.

2.2.2 Multi-band Analytical Techniques

There are a number of established techniques for exploiting spectral content of MS imagery for remote sensing applications (Tassel Cap [Kauth and Thomas, 1976], [Crist and Cicone, 1984]; Atmospherically Resistant Vegetation Index [Kaufman and Tanre, 1992]; Normalized Difference Vegetation Index [NDVI] [Goward, Markham, Dye, Dulaney, and Yang, 1991]; abductive polynomials [Drake, Kim, and Kim, 1993]; and principal component analysis (PCA) [Jiaju, 1988], [Cleva, Cachet, Cabrol-Bass, and Forrest, 1997], to name a few). However, when these techniques are applied to HS data, there are several difficulties. Many of these techniques are forms of band ratioing. NDVI, for instance, is simply a normalized ratioing of a near-IR band and a red band, which highlights healthy vegetation by exploiting a broad spectral feature of chlorophyll.

Determining useful band ratios in MS imagery is relatively simple. There are n*(n-1) possible ratios (n is the number of spectral bands of the sensor). We can assume that there are only half that number if we accept that inverse ratios highlight the same features with opposite contrast ($\frac{\text{band 1}}{\text{band 3}}$ is the inverse of $\frac{\text{band 3}}{\text{band 1}}$). For LandSat data (the thermal band is excluded because of its much larger spatial resolution), there are 15 (6*(6-1)/2) possible ratios. It is a simple matter to visually examine all 15 ratios to determine which generate useful contrast enhancements. It is even relatively simple to use these ratioed images as input for false color images and to visually investigate the false color images for useful representations. However, with 2,730 possible three-color combinations of ratioed images, it is unlikely all 2,730 would be examined exhaustively. Combinations based on the ratio images that highlight the features of interest would be used to guide the generation of false color images. However, for HS imagery, the number of combinations rapidly becomes unmanageable, as demonstrated in table 4.

One method of reducing the dimensionality of such a large search space is PCA and its variations (Smith, Johnson, and Adams, 1985; Lee, Woodyatt, and Berman, 1990; Harsanyi and Chang, 1994; Cowe and McNichol, 1985).

Table 4. Demonstration of the rapid growth in the number of images possible for HS and SP systems.

Sensor	Number of bands	Number of 3-band false color images using single bands	Number of unique ratio images	Number of 3-band false color images using ratio images
	n	n*(n–1)*(n–2)	m=n*(n–1)/2	m*(m–1)*(m–2)
Color TV	3	6	3	6
LandSat TM	6 (excluding thermal)	120	15	2 730
AOTF spectro-polarimetric single polarization	28 or 55	19 656 or 157 410	378 or 1 485	5.36×10^7 or 3.27×10^9
AOTF spectro-polarimetric two polarizations	110	1 294 920	5 995	1.66×10^{11}
AOTF spectro-polarimetric four polarizations	112	1 367 520	6 216	2.4×10^{11}
SEBASS	128	2 048 256	8 128	5.37×10^{11}
AVIRIS	224	11 089 344	24 976	1.56×10^{13}

PCA is a common tool used in multispectral imagery applications. The technique works best when knowledge of atmospheric conditions is available (for radiometric corrections of the data, before reduction). Otherwise, the atmosphere is mixed with the other components of interest, which makes it difficult to identify them. The knowledge needed for atmospheric compensation represents a significant effort in data correction, which at best requires corroborative measurements from field work and *in situ* instruments. Furthermore, the relevance of the corroborative measurements usually persists only for the time of the data collection; the measurements do not generally apply to any other times or locations. These types of measurements are not likely to be available during tactical operations, thereby reducing the usefulness of PCA type processing.

If a complete characterization of the atmosphere is not available for correction but some knowledge of the atmosphere is known, it is possible to use an atmospheric model such as ModTran or FASCODE (Air Force Research Laboratory, Hanscome Air Force Base, MA) to generate an approximation of the atmosphere. These models use the HiTran database of atmospheric absorption lines to generate models of atmospheric transmission, absorption, and radiance effects. These models can then be used in a number of atmospheric correction routines that can be applied to the sensor data (e.g., Richter, 1996, or Anderson, Pukall, Allred, Jeong, Hoke, Chetwynd, Golden, Berk, Bernstein, Richtsmeier, Acharya, and Ma, 1999).

These models are highly flexible and allow modeling of numerous conditions. Unfortunately, without knowledge of the actual values for many of the variables, any model computed will be approximate at best. At worst, it could severely corrupt the data. While determining these variables is usually easier than recording the actual atmospheric conditions, to generate a very accurate model requires a large number of variables and significant computation time. Further, these models are static and may quickly deviate from the real atmospheric conditions, so a model that corrects one image well may not do as well with the next image collected. These difficulties are often minor when one is working with MS data. These errors have a less significant impact because of the lower spectral resolution of the data. Thus, lower resolution models can be used, which require fewer parameters.

An alternative to using a model is to remove the atmospheric conditions from the data through an "in-scene" atmospheric correction. This approach uses the responses of pixels in the scene to estimate the effect of the atmosphere. To do this, there must be "known" materials in the image. The recorded spectrum of the material is compared to a laboratory spectrum of the material, the difference between them being the correction needed to account for the intervening atmosphere. Of course, this requires known materials to be in the image, a condition that will often not be true. Also, if the pixels used for correction are not purely of one material, then the correction will also remove the spectrum of the other material, which may not be the desired effect and generally reduces the usefulness of the correction.

2.2.3 Hyperspectral Techniques

These atmospheric corrections and models are of some value when they are used on HS data from sensors with large instantaneous FOVs (IFOVs). These systems, usually high altitude platforms, collect data where a single pixel has a footprint on the ground of a hundreds of meters or more on a side. Over this large spatial coverage, any small variations in the atmosphere are averaged spatially and thus "standard" atmospheric models can be used to correct the data to some extent. However, a tactical sensor may have an IFOV an order of magnitude or more smaller than an airborne or spaceborne platform (<1 meter square as opposed to >1 kilometer square). This removes the spatial averaging effect and as a result, corrections made with atmospheric models will often be significantly poorer on small IFOV systems than those made for systems with large IFOVs. The same goes for in-scene corrections, since the atmospheric conditions in one region of the scene can be very different from those in another region. These variations over the scene result in over- and under-corrections for many of the pixels, which will degrade detection and recognition performance.

Another source of error is that as spectral resolution increases, more atmospheric contaminants impact the system. Many of these contaminants are not included in the model calculations and many of them are not even in the absorption databases. Some of these contaminants have very narrow but strong transmission or absorption lines, which with lower spectral resolution systems are hardly noticeable but can have large impacts on higher resolution systems (see fig. 5).

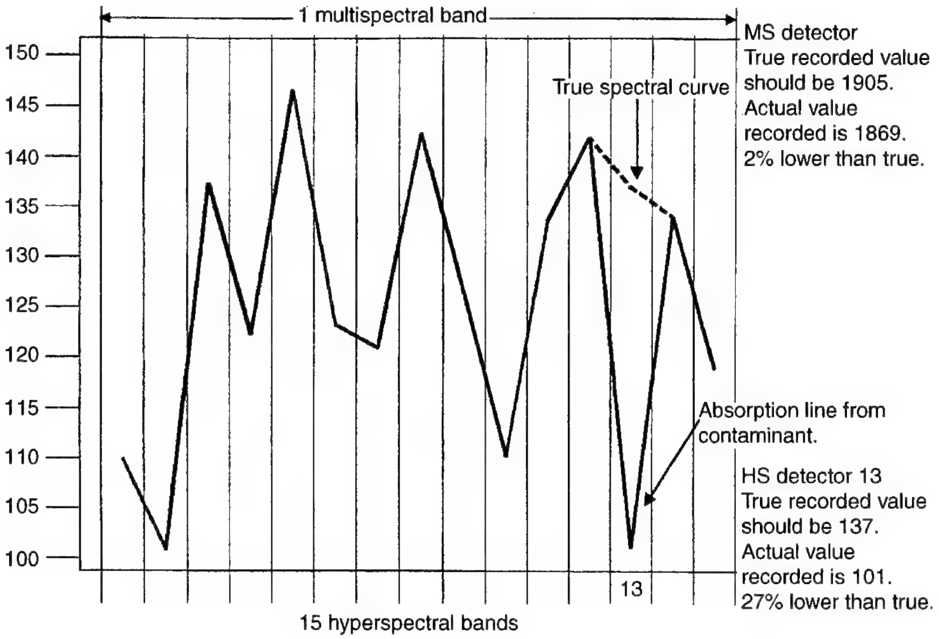
As HS systems improve in both spectral resolution and sensitivity, validation experiments comparing real atmospheric measurements to models of the same conditions continue to reveal deviations between the real and modeled results. Many of these discrepancies are attributable to approximations and simplifications made in the models, which were either ignored earlier because the error was slight or are only just becoming observable because of the improvements in the sensors. Significant ongoing research continues to improve and correct the models and the atmospheric corrections based on them.

Additionally, atmospheric correction techniques for HS data generally assume an airborne or space-borne platform that is looking straight down. For ground and low-altitude tactical scenarios, this will not be the case. The transmission path through the atmosphere in the tactical situation is very different from what is modeled for the overhead case.

For some situations, these deviations are not terribly detrimental. However, for high performance systems, such as those needed for a tactical system, these deviations, even when small, can severely impact performance. Achieving the high detection rates and low false alarm rates needed for a tactical system requires any corrections to be very accurate. Any correction that is not perfect will lead to some corruption of the corrected data.

The various difficulties with atmospheric models and corrections mentioned all combine to corrupt the "corrected" data and thus potentially reduce detection performance, increase false alarms, or both, which is usually

Figure 5. Why HS systems are more sensitive to contamination and noise effects.



unacceptable. As a result, it can be hit or miss as to whether a particular correction technique will perform adequately on a particular HS image. The need is for a more general approach, which is either less sensitive to these deviations or does not require corrections.

A large body of work exists in classifying high-resolution spectra under laboratory conditions. Using any electronic search technique will reveal papers, books, monographs, and journals about this topic, going back at least to the late 1800's. This type of data is generally not collected with imaging systems but can be thought of as a single pixel. These laboratory techniques are very successful in identifying material and chemical spectra. Some of these techniques are also successful in separating mixed spectra, that is, spectral signals containing spectra of two or more materials. The most common methods in modern laboratory spectroscopic analysis software are absolute value of differences, Euclidean distance, peak matching, least squares distance, first derivative least squares fitting and second derivative least squares fitting. These methods, with the exception of peak matching, all compare the query spectrum with a library of spectra and measure the differences among them. The unknown spectrum is identified as the library spectrum that it matches with the least error. The primary reason these well-established techniques are not more often used in remote sensing is their extreme sensitivity to the signal-to-noise ratio (Rauss, 1992). Studies have shown that just 5 percent random noise on a signal will significantly degrade performance of these techniques. During remote sensing conditions, the noise levels are often significantly higher than 5 percent. In some extreme applications, noise may be the dominant feature in the recorded data. The noise may be system noise as well as atmospheric contamination.

HS systems are far more susceptible to noise, contamination, and correction accuracy problems because of their narrow spectral bandwidths. An irradiance of acceptable strength for a MS system might be divided into 20 or 100 separate bands in an HS system. This would reduce the irradiance recorded by each band's detector by 20 or 100, respectively (see fig. 5). In remote sensing applications, the signal strength is often low to begin with because of low signal, fast sampling times, and atmospheric absorption; reducing it an order of magnitude or two further is a major concern for HS systems. It seems unlikely that HS remote sensing systems will often have signals strong enough for laboratory techniques to be applied.

Since 1990, the U.S. Geological Survey has been developing a system called Tetracorder (formerly known as Tricorder) (Clark and Swayze, 1995) for classifying HS data for remote sensing applications. Tetracorder replicates the process that an expert in spectroscopic analysis would follow using more than 120 high-quality laboratory spectra of environmental materials for comparison. Tetracorder uses a least squares curve-fitting method to compare the pixel spectra with the library spectra to find the best match. However, the system requires expert understanding and experience in spectroscopic analysis for proper use in any but the simplest of classification tasks. The

Tetracorder is very sensitive to the accuracy of the atmospheric correction, since comparisons of materials with similar spectral curves are greatly affected by inaccurate corrections. Furthermore, expert knowledge of the software and imaging system is needed to modify the Tetracorder software to match a new imaging system. Often, neither atmospheric data nor details about the imager are readily available with existing HS data sets. Significant effort also appears necessary to add new materials to the system library, limiting the method's usefulness in developing new or radical applications, especially since many of the contaminants and materials in a tactical scenario are not likely to be commonly seen elsewhere. As a result, it is highly unlikely that these materials are in the current Tetracorder library.

2.2.4 Spectro-polarimetric Techniques

SP processing, especially with respect to military applications, is a relatively new field. The spectral side of the data must address the same difficulties as the HS approach, which was discussed in the previous section. Polarimetery has been examined as a method of discriminating materials and surfaces in imagery. Reflected light's polarization depends on the three-dimensional geometry defined by the angle of incidence, the angle of reflection, the surface plane angle of the reflecting material, and the location of the observer, as well as the reflecting material's composition and smoothness. This is commonly described as the bi-directional reflectance distribution function (BRDF). These angles are highly variable and depend on the position and motion of the object, the observer, and the light source (the sun).

Smooth materials (often man made) tend to preferentially polarize reflected light, although to what extent depends highly on the BRDF geometry just mentioned. In the IR, structural materials (metal, glass, plastic) and coatings (paints) appear smooth, resulting in observable polarization effects.[4] However, because of the variability of the BRDF, polarization may be an inconsistent feature. Polarization is also greatly affected by the atmospheric conditions, especially aerosol and particulate concentration, which can severely alter the polarization of the light as it travels to the sensor. This could mask the polarization differences that could be used for detection and discrimination tasks. It seems unlikely that material will be identifiable by polarization alone. However, it may be useful in clutter suppression and possibly in monitoring atmospheric conditions.

Clearly, new techniques are needed to address at least some of these issues in order to take advantage of the information available from HS and SP sensors. The remainder of this report discusses experiments that employ an adaptive learning technique called genetic programming for detection and recognition of different materials in HS and SP imagery.

[4] See IRIA technote: http://csdnta.erim-int.com/IRIA/ SRPUB.NSF/ 957aa8092ef9753f85256747004f9331/200236e47983509f852563ed004e0969?OpenDocument

3. Genetic Programming

GP is a method of automatically generating a computer program (or, as used in this particular work, a mathematical equation) to solve a particular problem. GP is a branch of genetic algorithm (GA) research. While GP shares many of GA's underpinnings, terminology, and operators, GP's implementation is quite different, resulting in substantially different results than the GA approach. GP accomplishes this through a series of operations similar to biological evolution. These operations are reproduction, crossover, and mutation. A number of good books about the topic are available, including Koza, Bennett, Adre, and Keane (1999) and Banzhaf, Nordin, Keller, and Francone (1998), as well as several web sites. Two good internet tutorial sites are www.genetic programming.com/Tutorial/index.html and www.genetic-programming. com/gpanimatedtutorial.html.

The GP technique used in this work operates on equation trees. The result of the GP learning process is a single mathematical equation that obtains the most correct answers with the training data. This equation uses simple mathematical operators to combine the intensity values of the individual HS bands to compute a single numeric value for each pixel. This value determines the pixel's classification as the desired class or not. Obviously, given some constraints on the set of equation trees, such as available operators (mathematical functions) and terminals (the band intensities and numeric constants), the maximum and minimum allowable tree depth, and the maximum allowable tree breadth, one or more equation trees will obtain a minimum error with the training data. An obvious way to find this equation(s) is to process every possible equation in this set. Unfortunately, as the number of terminals and operators increases and the tree breadth and depth increase, the number of possible equations becomes impossible to fully search in a useful time frame with currently available computer systems.

What GP does is allow a stochastic searching of this equation space through evolutionary development of the equations. The GP process works like this. A large (but manageable) number of random equation trees are generated. This initial set of equations is referred to as the "generation zero population". These equations are evaluated with the training data against some fitness function. For this effort, the fitness function is the number of correctly classified training pixels. How this is done is covered in section 4.6.2. The best (most accurate) performers are then used to create some number of new equations to replace some of the poorer performers. This new population of equations is the next generation: Generation 1.

The new equations in the population of Generation 1 are obtained by the use of one of the three evolutionary operations on the best performers of Generation 0. Which operation used is chosen probabilistically. In one possible operation, a "best performer" (BP) has a probability of being reproduced exactly in the Generation 1 population, thus increasing its odds of being selected in future generation if it continues to perform well. In the second possible operation, each BP has a probability that it will be mutated. The mutated equation is then added to the Generation 1 population. There are two types of possible mutations. In the simplest, one node of the tree is randomly selected and randomly changed (following any constraints of that particular node; branches cannot be replaced with leaves, etc). Figure 6 shows an example of two simple mutations, one on an operator and one on a variable. Only one of these types of mutation would be used to create a new equation; they are combined in this figure only to save space.

The second type of mutation is replacing a branch in the tree with a copy of another branch. Figure 7 shows an example of this type of mutation.

In the third possible operation, a selected BP equation tree is "bred" with another BP equation through a cross-over operation. In a crossover, the two original equations, called the parents, swap randomly selected branches, creating two new equations, the children. Figure 8 shows an example of crossover.

Figure 8 also shows how a common formula could be evolved through the GP process. The child on the right is the tree representation of the quadratic equation. This type of tree crossover differs from a GA implementation in

Figure 6. Example of mutating an operator and a terminal. (Note: only one mutation would be performed in any one operation.)

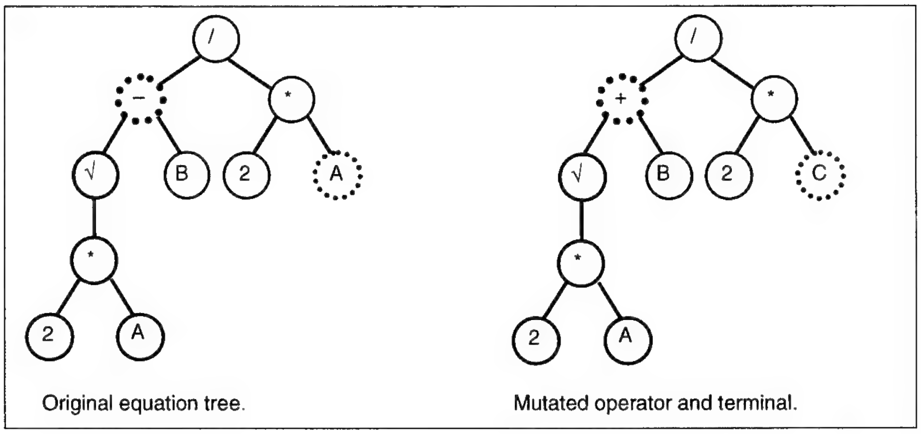


Figure 7. Mutation of tree through replacement of one branch with a copy of another. (The bold sub-tree in the left tree is replaced by a copy of the outlined sub-tree.)

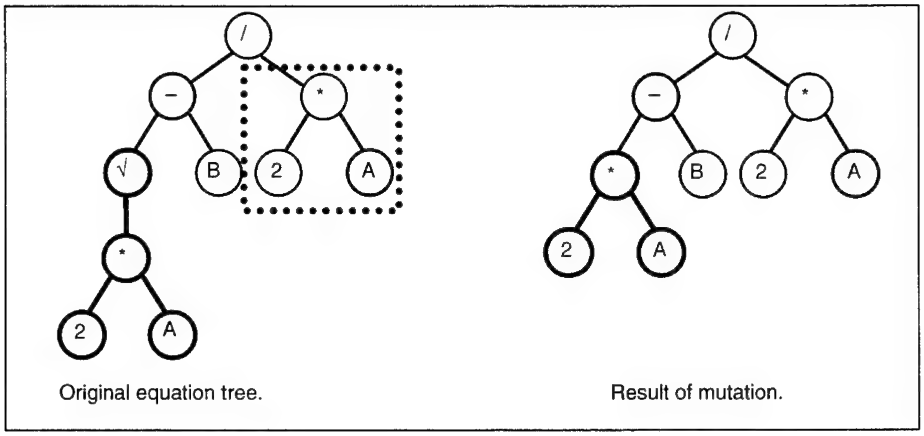
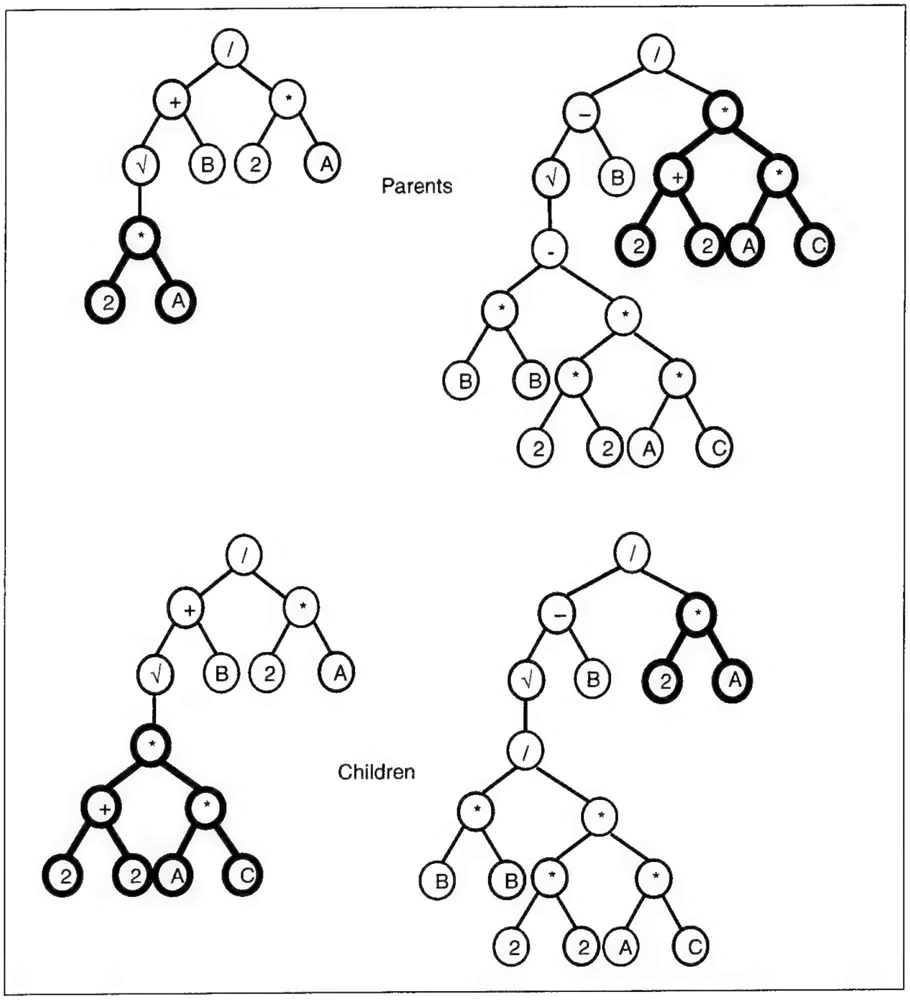


Figure 8. Cross-over operation. (Bold branches are swapped between parents to create new children.)



that if the same tree is used as both parents in GA, both children are identical, whereas in GP, the children are different, unless the branches swapped happen to be identical.

After this "breeding" phase, the new population of equations is evaluated against the training data. Then the Generation 1 "best performers" are used to create new equations for the Generation 2 population of equations. The Generation 2 equations are evaluated and so on, until an equation correctly classifies all the training data or until some pre-defined number of generations is created. The "best" equation from this process is a solution to the problem.

This equation very likely will not be the best possible solution. It is only the best equation found through evolving the equations in the Generation 0 starting population. It is important to remember this because beginning with a different starting population will usually lead to a different solution equation unless the problem to solve is very simple.

To find a more optimal solution, it is necessary to run the GP learning process many times, beginning with different starting populations each time. Performing the process many times allows a stochastic sampling of the entire space of possible equations. Once the equation space has been adequately sampled, the "best" equations from each run are collected and then compared to determine which are the most useful for the task at hand.

3.1 GP in Image Processing

Encouraging results have been obtained with GP techniques used with other types of remote sensing data, such as synthetic aperture radar (Daida, Hommes, Bersano-Begey, Ross, and Vesecky, 1996; Daida, Onstott, Bersano-Begey, Ross, and Vesecky, 1996) and infrared line scanner data (Roberts and Howard, 1999). GP has also been shown to be a key tool in image processing and feature extraction (Tackett, 1993; Brumby et al., 1999; Howard and Roberts, 1999).

A key advantage of GP versus other adaptive learning techniques is that GP gives an explicit equation as its result. For example, take a simple hypothetical MS example that uses LandSat TM data. Assuming that the desired result is to contrast vegetation pixels from other types of pixels, the GP process could result in the following equation:

$$\frac{\text{band } 3 - \text{band } 4}{\text{band } 4 + \text{band } 3}$$

which is a version of the standard NDVI equation used for LandSat TM data.

This example shows how GP could be used for data mining or compression. Assuming that the highlighting of green vegetation is the goal, the equation shows that only two of the six bands are needed. This may indicate a method of reducing or compressing the data for transmission purposes. For this example, only Bands 3 and 4 are being used. If highlighting vegetation is the only task the system is to perform, then only these two bands need be transmitted from the system. The equation itself may also expose some relationship between the input values that may be exploited with other techniques. In the example, the equation is exploiting a significant spectral feature of chlorophyll.

Adaptive learning techniques such as GP are attractive for HS image processing because such techniques adaptively "learn" any necessary radiometric and atmospheric corrections. The user does not have to explicitly state (or even be aware of) those corrections. Furthermore, these techniques can also address spectral contamination, such as pollutants in the atmosphere. None of the more common techniques even try to address spectral contamination because it is impossible to adequately model all the possible conditions. Furthermore, atmospheric contaminants that may commonly be seen on the battlefield are not commonly found in the natural environment, which is the primary focus of most remote sensing work.

Using GP on HS data has the potential to address many of the difficulties of working with HS data. This method could be a relatively simple tool to allow application design and experimentation. This could encourage the development of new applications. The method is supplied with all the band information during training, allowing the exploitation of the finer spectral resolution of the data. Spectroscopic techniques are not required to design classification experiments. If a poor choice of classes is made, GP will fail to develop a satisfactory solution. This tool could be relatively easy to use. It would not require expert spectroscopic knowledge for useful applications to be developed. Finally, GP results in a specific classification equation. This equation can be evaluated to try to understand the underlying physics of the classification and to reduce the storage capacity needed for application data. One can reduced storage by determining the spectral bands necessary for the particular classifications and by allowing the use of smaller subsets of the data to reside in active storage.

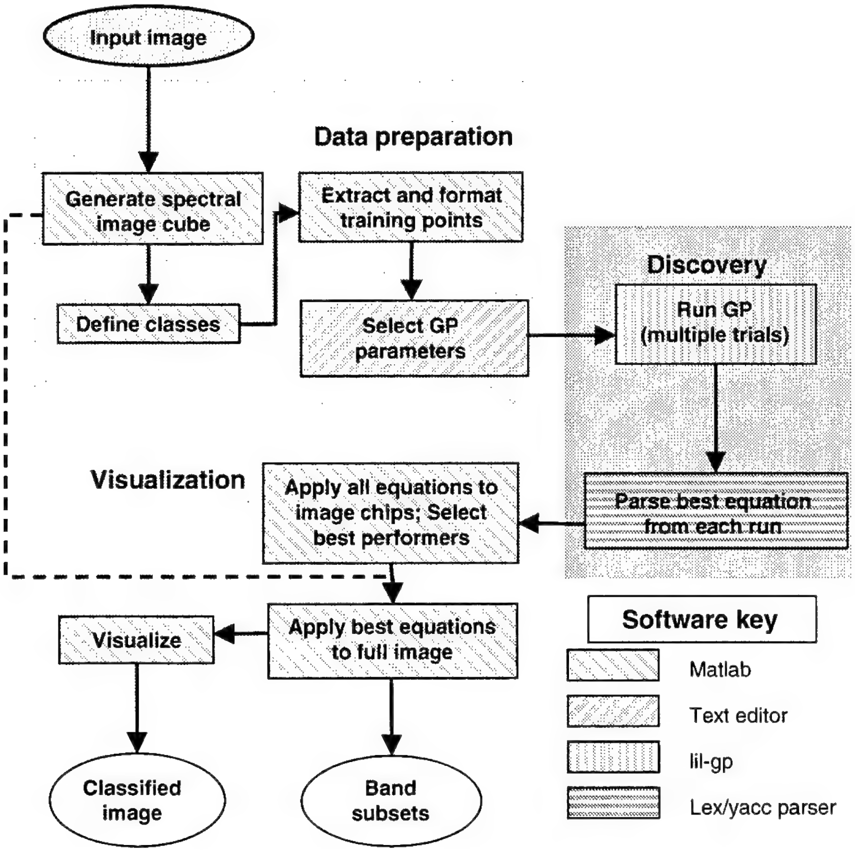
4. GP for Hyperspectral and Spectro-Polarimetric Data

The goal of this research project was to construct a computer-assisted design system for the discovery of classification algorithms and germane band information, via a small number of examples of sensor data from user-defined classes. This system is built around genetic programming to automatically develop classification algorithms that will perform pixel-level spectral classifications of HS data.

4.1 System Design

Daida, Hommes, et al. (1996), Daida, Onstott, et al. (1996), Daida, Bertram, Polito, and Stanhope (1999), Daida, Polito, Stanhope, Bertram, Khoo, Chaudhary, and Chaudhri (2000) have been using a GP software package called "lil-gp" (Zongker and Punch, 1995) for other GP research efforts. Daida's research group has made several modifications in it (Daida et al., 1999). This modified version of "lil-gp" is the core of the system reported here. I had previously developed a number of MATLAB® scripts and functions for the manipulation of HS image cubes. Together, these were used to create the system design in figure 9, an early version of which was presented in Rauss, Daida, and Chaudhary (2000).

Figure 9. Data flow of current system.



The goal of this project was to demonstrate two things: first, that it is possible to use GP techniques to classify pixels in a spectral or SP image. Second, it is potentially useful to do so. We planned to show that this could be done by using as many existing software tools and modules as possible. There was little point in designing a streamlined, user-friendly, optimized system from scratch if it was not going to be useful. Following the data flow in figure 9, the system's modules are now described.

4.2 Imagery Used

Obviously, before one can begin to even design such a system, some data must be available to evaluate each module and the complete system. The data used for this effort are from an experimental SP sensor developed under the auspices of the Federated Laboratories Consortium by Carnegie Mellon Research Institute (Denes, Gottlieb, Kaminsteyi, and Huber, 1997). The imagery was collected for the U.S. Army Research Laboratory (ARL) (Gupta, Dahmani, Gottlieb, Denes, Kaminsky, and Metes, 1999) in 1998. This system (hereafter referred to as the acousto-optical tunable filter [AOTF]) has a spectral range from 0.46 to 1 micron and has a tunable spectral resolution, usually 10 or 20 nm. The system uses an AOTF to create narrow band filters to capture snap shots of a scene with a charge coupled device camera. We can select both the bandwidth and center wavelength of the filter by adjusting the acoustic frequency input to the filter. This system also includes a phase retarder that is used as a polarization filter. This allows the collection of polarization information for each spectral band.

The imagery used for this work is spectro-polarimetric, with 28 spectral bands and 4 polarizations for each band: vertical (0 degrees), horizontal (90 degrees), and ±45 degrees. This gives the full cube 112 bands. Four image cubes with the same spectral and polarization characteristics were identified from the data available. One image cube was used as the cube from which training data were extracted. While none of the images are well ground truthed, it is possible to clearly identify five distinct classes in the training image, from which training pixels could be extracted. Figure 10 shows a single band of the training image from this system.

4.3 Spectral Image Cube

After the training data to use were identified, it was necessary to assemble the various tools to allow the visualization of the data. As figure 9 shows, most of the tools were written in MATLAB®. The AOTF data had been recorded as single .bmp files for each polarization and spectral band. Using MATLAB's image processing toolbox, it was simple to use the imread() function to automatically read the .bmp files into a three-dimensional MATLAB® variable. This variable is the spectral image cube. The final cube size is 480 by 640 by 112 (y dimension by x dimension by spectral bands). This single three-dimensional MATLAB® variable was then saved and used for the remainder of the project. The .bmp files could then be removed from the active disk area. The same was done for the other three evaluation images.

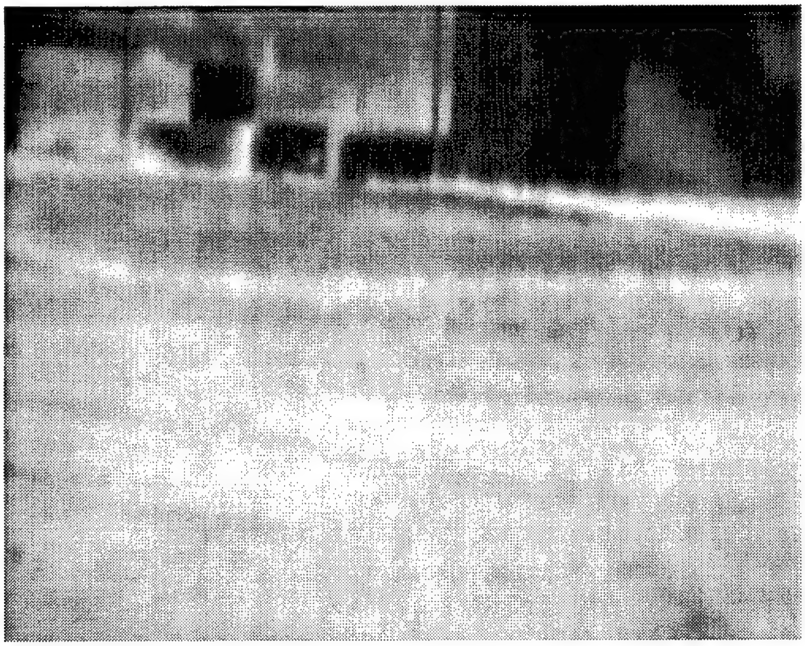


Figure 10. Band 72 of AOTF image from which the training pixels were extracted.

4.4 Target Classes

The next step in the process is definition of the classes to be learned by GP. Class definition is one of the most critical steps in the process. Identifying a single spectrum as more than a single class would make it impossible to "train" any supervised learning system. For the AOTF data, five classes were identified: *grass, pad, wall, barrel bright*, and *barrel dark*. These five classes are rather obvious. I was reasonably confident that these classes could be learned because I had previously trained a neural network with the same training data, which had little trouble learning to separate these classes.

Figure 11 shows Band 76 of the SP image cube. The circles and x's depict the approximate location of the ten pixels extracted for training spectra for two classes. The x's represent the 10 pixels chosen as the class *grass*. The circles represent the class labeled *pad*. While there is no ground truth available for these data, I believe the pad is an area of bare earth, perhaps a parking area in front of the building.

Figure 12a shows an enlargement of the left side of the building. This area is where the ten pixels of the class wall were extracted. The crosses show the approximate location of the pixels used for training.

Figure 12b shows an enlargement of the barrel. The upper dark portion of the barrel was defined as the class barrel dark. The bright pixels show the approximate location of the ten pixels extracted for the training set for this class. The lower section of the barrel is defined as the class barrel bright. The black pixels in this area show the approximate location of the ten pixels chosen for this class.

Figure 11. Single band of AOTF spectral image showing approximate areas where training pixels for *grass* (X) and *pad* (circle) classes were extracted. (Note: image is 640 by 480 pixels in size.)

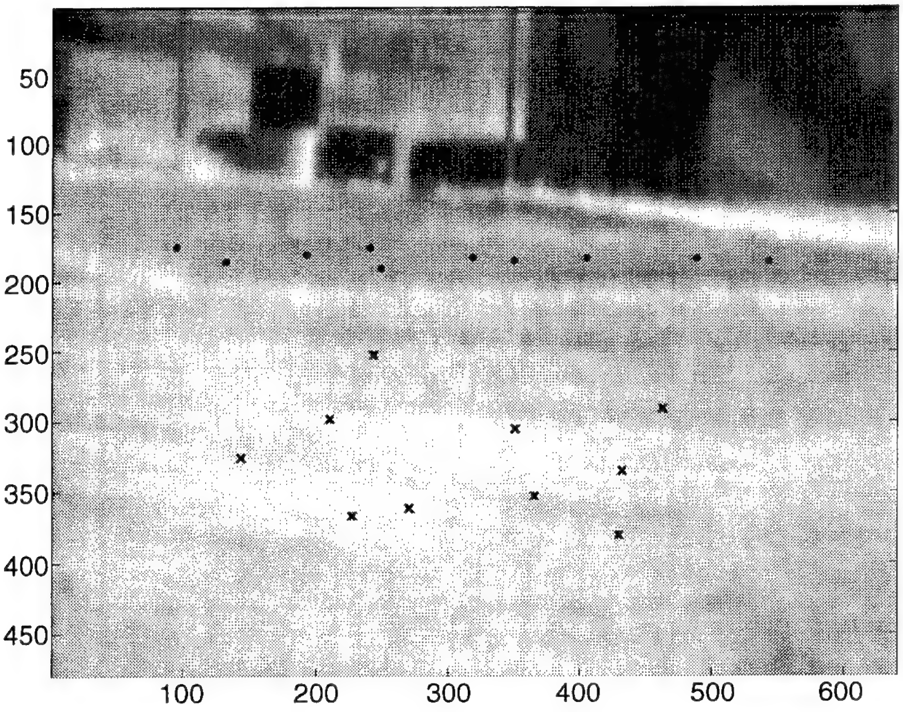
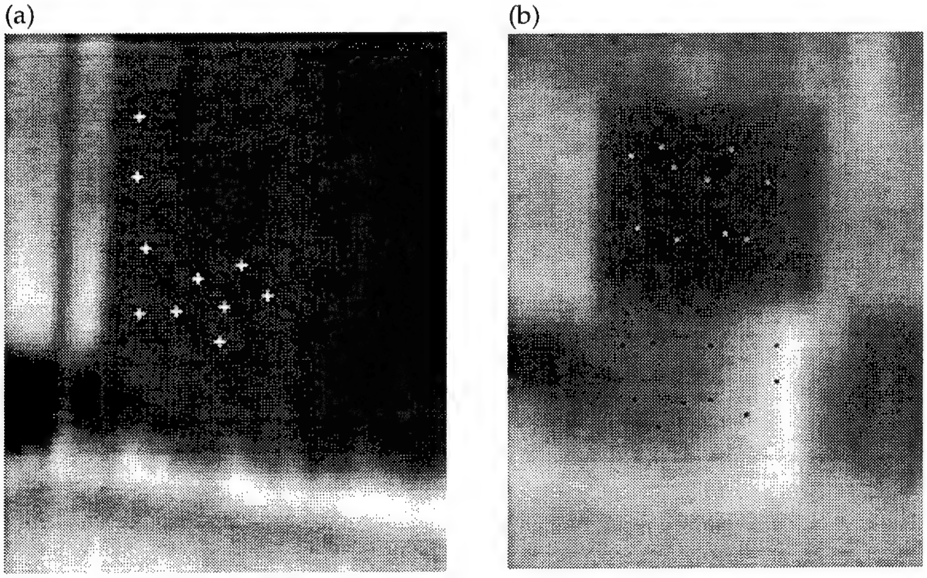


Figure 12. a and b: Enlargements of Wall and Barrel with approximate location of training pixels marked.



4.5 Training Pixel Extraction

I had previously written a MATLAB® script that loaded a spectral image cube and displayed a simulated broadband image, created by summing each pixel over all the bands. Using this image, the user can zoom in on areas to allow selection of individual pixels. The user defines a class number and clicks on as many pixels as is desired for a training sample. The locations of these pixels are then used to extract the spectra of each pixel and store them in a MATLAB® 2-D variable for each class. Figure 13 shows the spectra of 20 random pixels from each class. Ten of these from each class were used as the training pixels.

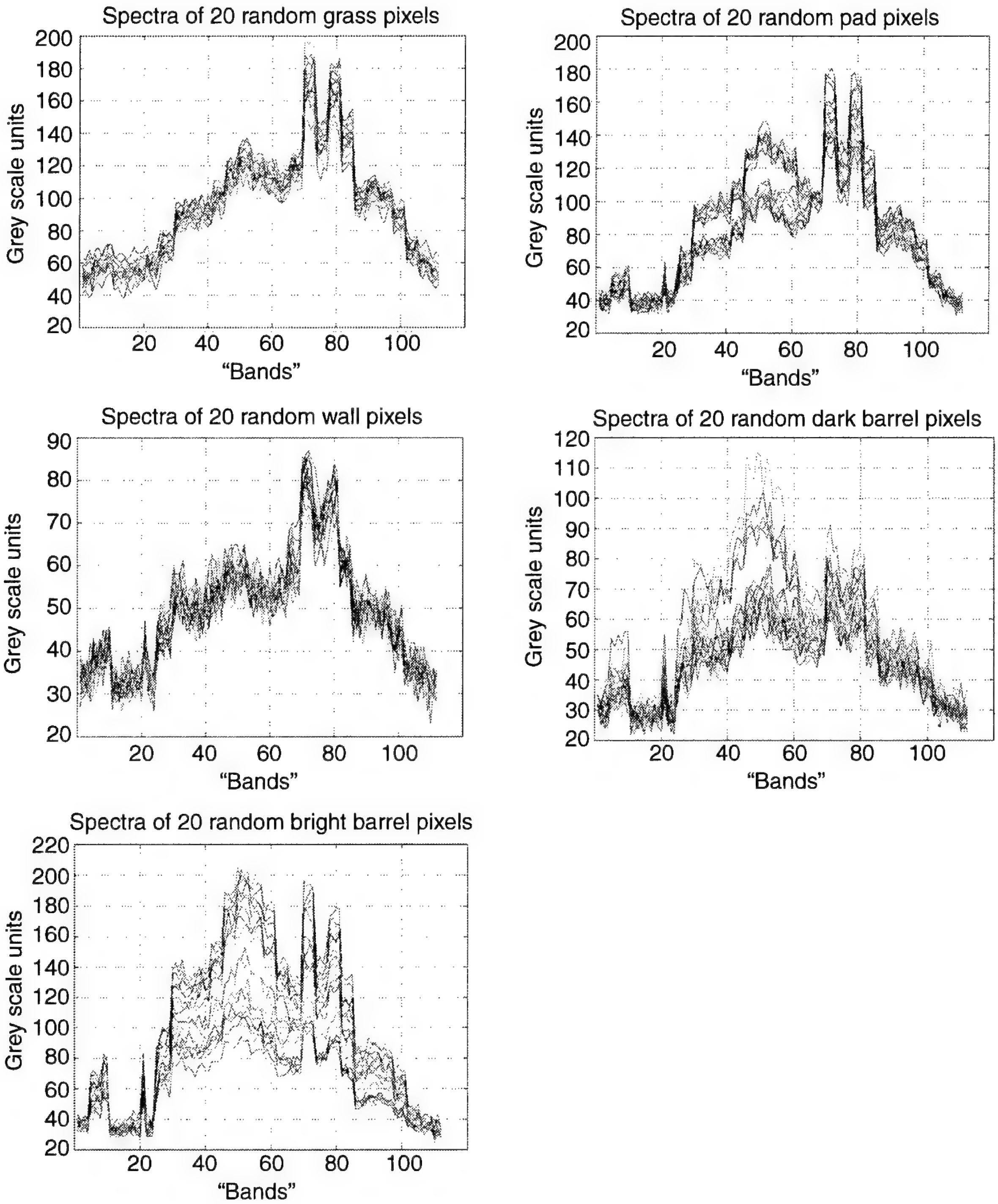


Figure 13. Raw intensity values of 20 pixels from each class. (Ten from each class were used as training data.)

After extraction of all the training pixels desired for each class, another MATLAB® function is called. In order to remove intensity information, it takes the class variables and normalizes each spectrum individually to a range of –1 to 1. Normalization of individual pixel spectra to a range of –1 to 1 removes the possibility of the GP using intensity as a discriminant (see fig. 14).

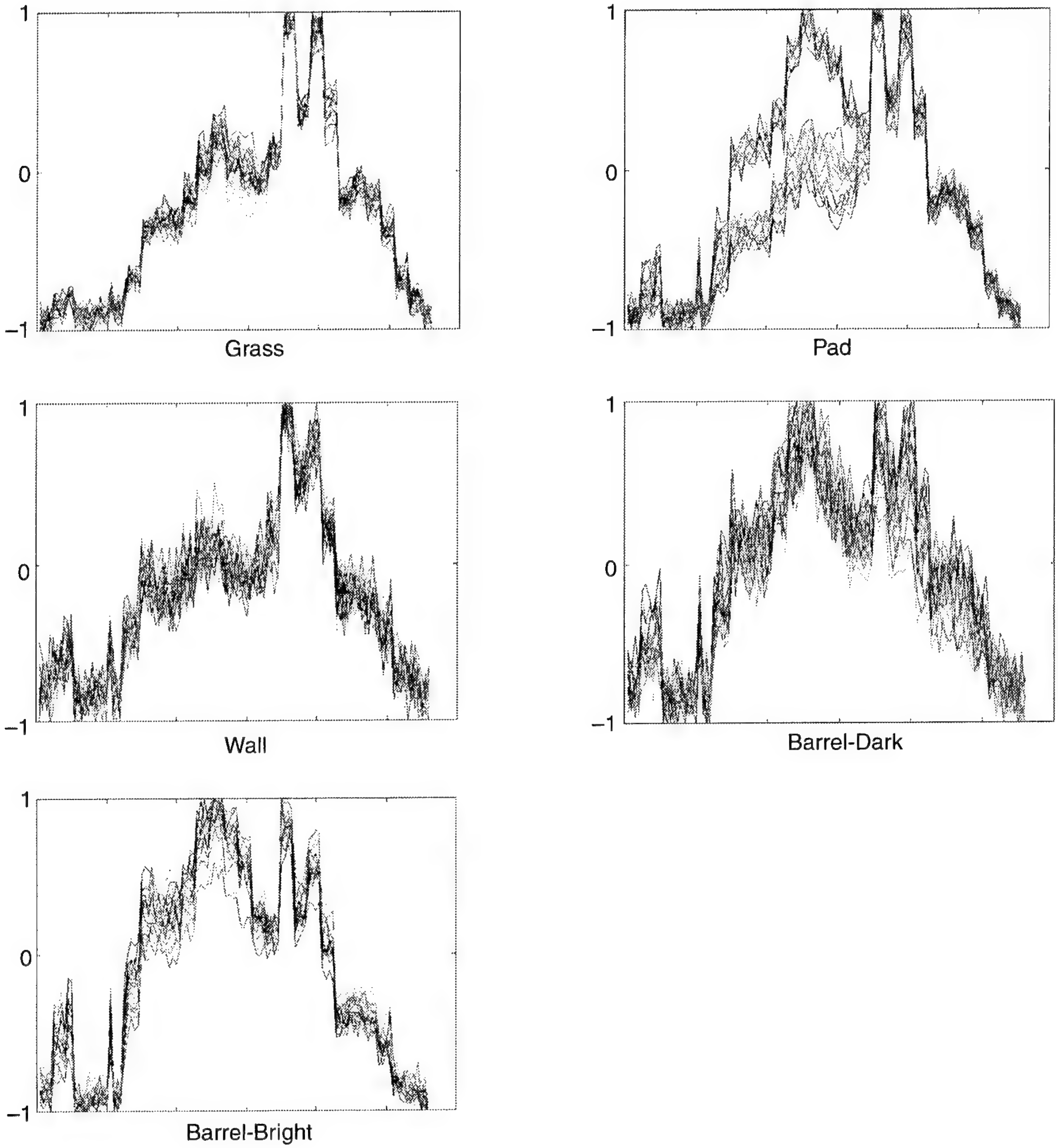


Figure 14. Normalized training spectra for each class.

Intensity will vary greatly, depending on the changing illumination. For example, in figure 12b, the left side of the barrel appears to be in the shadow. With intensity values in this image, it would appear that the left side of the barrel is different from the right side.

However, this is just an effect of the illumination, which appears to be coming from the right side of the image. The spectra of the two paints on the barrel will not change—only the intensity of the curves change (see fig. 14).

Normalization removes the task of discovering that intensity is not a useful feature and should result in a more generalized classification equation. These results can then be used on a broader set of data. As the function normalizes the data, it writes the spectral vector to a text file, which can be read into the GP software.

4.6 GP Software and Parameter Selection

4.6.1 The GP Software

The GP portion of the system was built around the lil-gp software package (Zongker and Punch, 1995). The version used for this effort is a patched version (Daida et al., 1999). At the start of this project, lil-gp only recognized a single terminal (variable) x. Lil-gp could only generate equations such as $(x+x+3)*4/x$. To use lil-gp on multiband imagery, it was necessary to expand lil-gp's capability to accept a large number of variables—in this case, the intensity values for each band in the data. Shahbaz Chaudhary made the necessary modifications in lil-gp to recognize multiple, independent variables (the bands). To avoid modifying the core code of lil-gp, the capability to recognize multiple variables was added through modifications of the user-defined functions that lil-gp uses. As a result of this modification, lil-gp could now generate equations with multiple variables such as

(*(/(+(+ b1 b2) b3)(-(+ b4 b5) b6))(/ b7 b7))

in which b# represents the band number the value is taken from, b1 for band 1, b7 for band 7, and so on.

To limit the amount of initial development required with lil-gp, only the simplest of operators: +, –, *, and protected division (/) were used. Protected division differs from normal division in that division by zero returns a one.

The lil-gp software is flexible enough that additional mathematical operators may be added with minor effort. Furthermore, logical operators may also be implemented (see the lil-gp documentation for details [Zongker and Punch, 1996]). These additional operators may be useful for HS image processing but were beyond the scope of this investigation.

4.6.2 Parameter Selection

Lil-gp reads an input text file that contains the parameters used for initialization and processing control. Most of the parameters used are similar to those mentioned in Chapter 7 of Genetic Programming: On the Programming of Computers by Means of Natural Selection (Koza, 1992): population size = 100, crossover rate = 0.9, replication rate = 0.1, population initialization with ramped half and half, initialization depth of 2 to 6 levels, and fitness proportionate selection. Other parameter values were maximum generations = 100 and maximum tree depth = 20. "Population size" is the number of equations to be randomly generated for the starting population.

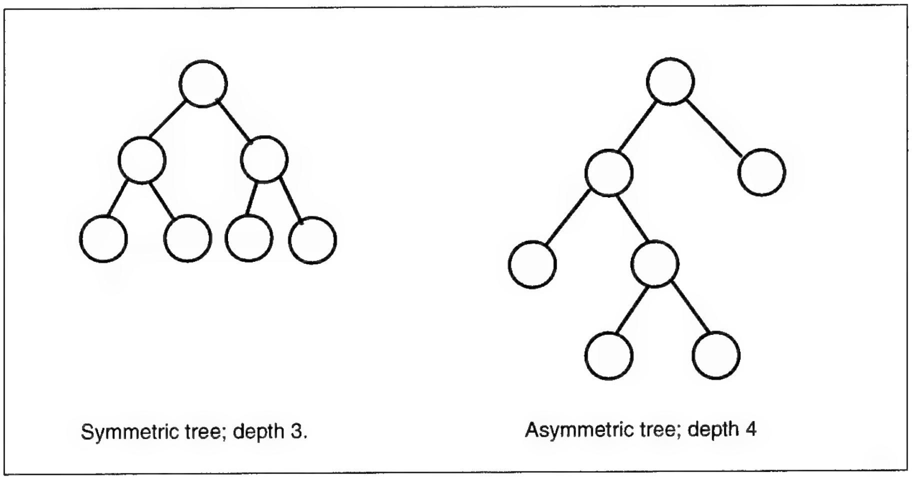
The "cross-over rate" and "replication rate" define the probability that a selected equation tree will be bred or replicated. Note that the mutation operation is not used in the work reported here.

The "population initialization" parameter "ramped half and half" means that during the generation of the initial population of equation trees, approximately half of the time the tree will be built in a balanced manner so that the resulting equation tree is symmetric. The other half of the time, the tree will be grown by augmenting randomly selected nodes, which often results in asymmetric equation trees (see fig. 15). The "initialization depth" of 2 to 6 indicates that during the initial population generation, a tree has a 20 percent chance of being built to a depth of 2, 20 percent to a depth of 3, and so on. Therefore, the initial population will have a relatively even distribution of trees from depth 2 to 6. The distribution is only relatively even because there are conditions when tree growth may be stopped before it reaches the target depth. Also note that duplicate trees are prohibited.

"Fitness proportionate selection" means that the probability that a given equation will be selected for breeding or copying into the next generation is proportional to its "fitness," how well it solves the given problem. The "maximum generations" of 100 indicates that only 100 generations of equations will be evolved. If a 100 percent correct solution is not developed by this time, the run is to be stopped. Finally, the "maximum tree depth" is the largest tree depth allowable for any evolved equation.

With any text editor, modifications of the parameter file of lil-gp can be made. "Population size" (the number of equations in a population) varied as the system developed. Early evaluation runs were done with a value of 100; later runs, reported in section 5.4, used a population size of 10,000. Other settings that were changed during development were the "maximum tree depth" and the "selection method." Maximum tree depth was reduced from 20 to 15 during later runs based on unpublished (to date) results from other research in Daida's research group, which indicated that a tree depth of 20 may be too large to produce useful results. Daida's work seems to

Figure 15. Example of symmetric and asymmetric trees.



show that GP applied to other (albeit smaller) problems rarely finds useful solutions beyond a tree depth of 10 to 15. The fitness selection method was also changed from "fitness proportionate" to "fitness overselect." For large population sizes, a "greedy overselect" method is recommended. In this method, the population is partitioned into two groups. The separation of the populations is done by ordering the trees according to decreasing accuracy and then selecting the top X% of the equations as Group 1. The remaining trees are Group 2. X is based on the population size. The default setting is 320/population size. Based on this separation, 80 percent of the time, a tree is selected from Group 1. Once a group is selected, the specific tree selected is chosen in a fitness proportional manner.

The input file also contains the starting random seed and the target class for identification. The random seed defines the resulting content of the starting population. With all other settings remaining the same, a change in the random seed will result in a different starting population. Thus, by running lil-gp many times and changing the random seed each time, we make a stochastic sampling of the entire set of possible equations. Many more alternate settings may be used (see the lil-gp user's manual [Zongker et al., 1996] for full details).

In order to assess the usefulness of an equation, the equation's response is evaluated with some type of fitness function. The fitness function used in this work is very simple. It was designed to separate one class from the other four. The choice of the "desired" class is made selectable through the use of a variable in the parameter definition file. The actual fitness functions used are discussed later in section 5.4. By defining which of the five classes to use as the "target" class in the input file, it is relatively easy to maintain all the other settings the same and obtain results for the different classes (see the next sub-section about batch runs). The fitness function is currently two "if statements," one checking if a training pixel defined as the target class returns a response in the appropriate range and the other checking if a training pixel from any of the non-target classes returns a response in the appropriate range. The fitness function is relatively easy to modify and can be as sophisticated or as simple as one wants. Initial development used a simple zero threshold. It checked whether the target class responded with a value >0 and non-target classes with a value <0. Several other fitness functions were used and are discussed in section 5.4. The number of training pixels correctly classified defines the performance of an equation.

4.6.3 Batch Runs

Lil-gp is restricted in the size of the initial population of equations it can generate because of computational and memory constraints. Furthermore, it only reports the first equation that obtains the best results. Other, equally successful equations may exist in its population, but they are not reported in the current implementation of the lil-gp software. For these reasons, lil-gp needs to be run repeatedly, each time with a different random number seed for initialization.

To automate the process of running lil-gp with different random seeds, a "perl" script used by Daida's group in an earlier project was modified. This script performs two tasks. First, it cycles through each of the five classes as the "desired target" for a given random seed, resulting in a solution equation for each class. Second, the script loops through a series of random seeds over a given range. This allows an experiment to be set up to generate equations to separate each of the five classes from the other four over a range of different starting populations, without further monitoring by the user. This is useful as even the small number of runs done for this work (20 to 23 different random seeds for each class) took almost 48 hours to complete. While working on a problem with data sets this large is uncommon for GP at this time, it is believed that to obtain an adequate sampling of the equation space for a nearly optimal solution, hundreds of different random seeds must be used and the results of each must be evaluated.

4.7 GP Output

Lil-gp generates a number of result files. The only one of significance to this report is the .bst file. This file contains the first "best" equation from the current generation. Figure 16 shows the contents of a .bst file for an early single polarization run; only 28 spectral bands of a single polarization were used.

This text file contains other useful information besides the equation. Generation is the generation in which this equation first appeared. Nodes is the number of nodes in the tree. In the example, the value is five. Looking at the last line in the example, these five nodes are –, –, b27, b8, and b9. Depth is the depth of the tree. Hits is the number of correct answers that this equation gave for the training data. Raw fitness is just the number of correct hits. Standardized fitness is the number of incorrectly identified training values. Adjusted fitness is 1/(1+standardized fitness), an artifact from an earlier implementation of lil-gp.

Figure 16. Example of a simple .bst file from a lil-gp run.

```
=== BEST-OF-RUN ===
              generation: 14
                   nodes: 5
                   depth: 2
                    hits: 47
TOP INDIVIDUAL:
-- #1 --
                    hits: 47
             raw fitness: 47.0000
    standardized fitness: 3.0000
        adjusted fitness: 0.2500
TREE:
 (- (- b27 b8) b9)
```

4.8 The Parser

Lil-gp produces its best equation tree in a operator-lhs-rhs (prefix) style, resulting in equations that look like this test equation:

$$(*(/(+(+\ b1\ b2)\ b3)(-(+\ b4\ b5)\ b6))(/\ b7\ b7)) \tag{1}$$

in which b stands for band.

MATLAB® is used to process imagery with a GP equation. MATLAB®, however, requires a standard mathematical (infix) format (lhs-operator-rhs). Again, fortunately, a parser was available form earlier work by Daida's group (Daida et al., 2000). The parser is built with "lex" and "yacc" (standard programs common in most UNIX C programming packages). Lex is used to construct a token generator to read the equation, extract tokens, and execute specific C instructions upon finding certain tokens. The lex code was designed to evaluate and simplify instances of *1, /1, +0, and –0, and to check for instances of var/0, which is replaced with 1, since protected division is used. Lex generates a lexical analyzer for use with yacc. Yacc creates the program that performs the actual parsing and produces the reordered (and possibly simplified) equation. This results in equation 1 being parsed and transformed into

$$((((b1)+(b2))+(b3))/(((b4)+(b5))-(b6))) \tag{2}$$

The parser was designed to parse files generated from batch runs in a single call. The parser saves the parsed results in a single text file. As a result, by using the proper file name to identify the target class, the parser generates a single text file with all the parsed equations for the target class. The parser was modified to accept multiple variables, since its original implementation, like lil-gp, was designed for only single variable problems.

A useful feature of the parser is that the number of correct responses (hits) that an equation gives for the training data can be used as an acceptance criterion for parsing an equation. For this effort, the value was arbitrarily set to 45. Forty-five of the 50 training pixels must be correctly identified by the GP equation for it to be processed and put in the parsed equation file. Equations with fewer than 45 hits will not be parsed. This culls the poor performers from the parsed result file.

4.9 Evaluate Batch Results

The resulting equations from the parser can then be copied into a MATLAB® script designed to process three small chips from the AOTF training image. The use of chips to compare equation performance is the most recent addition to the system data flow. Since processing a complete AOTF image may take well over an hour per equation, it became obvious that for batch runs of the size needed to identify optimal solutions, it would take days to run the image through every equation obtained. It would be very inefficient to process the data this way, as only a small percentage of the equations are expected to perform at an acceptable level.

The chips are spectral image cube subsets of the full AOTF training image. Each of the five classes appears in at least one of the three chips. The chips were chosen to be small enough to greatly accelerate the processing but large enough for one to visually evaluate each equation's performance in separating the classes.

Three image chip files are used to pre-screen the equations for usefulness (see fig. 17). The results reported here come from batch runs of 20 to 23 trials. To process the complete image for a single class for all 20 equations would take approximately 20 hours to complete. Using the chips, all 20 equations take only 40 to 60 minutes to process. Displaying the resulting classification images for each equation, it was possible to visually determine which equations had the most potential. The number of useful equations for a class separation varied from two (rather poor ones) to nine. The longest complete image run for nine equations of a single class took just under 7 hours.

Note that MATLAB®, being an interpreted computer language, is not very fast. Should large runs be necessary, it may prove valuable to move to a compiled language such as C or C++, which would greatly increase the speed of processing.

These image classification chips, saved as MATLAB® multidimensional variables, can then be displayed and evaluated. This resulted in a collection of five chip sets, one for each class separation. Appendix A shows two examples for each class. These classification chips were displayed and scored subjectively as to their success in highlighting the desired class and suppressing the other four classes.

While there may be bounds on the acceptable returned value from an equation during training, there are no bounds on the responses of the equations during the processing of images and chips. As a result, there are often some very extreme values in the resulting classification images, which makes visualization of the raw results difficult. To generate a useful visualization of the images and chips, an adjustment in the gray scale of the image is made. For all the classification images and chips shown here, the gray scale was set to saturate to black at a value of –0.01 and saturate to white at a value of 0.01, essentially creating a binary image thresholded at zero.

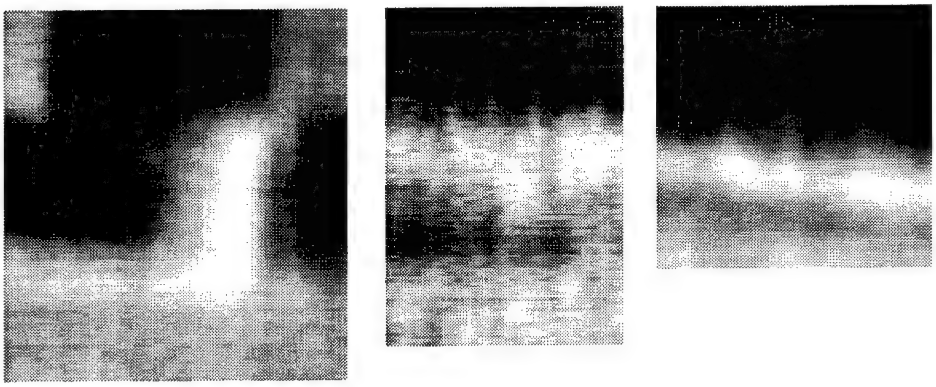


Figure 17. Image chips used to evaluate batch-generated equation performance. (Left chip contains both barrel classes, with some grass and pad. Center chip contains grass and pad. Right chip contains wall, grass and pad.)

The two "best" wall equations demonstrate the point that the 20 runs performed may not have adequately sampled the equation space. These two equations are obviously not very good with many false positives and false negative responses.

4.9.1 Protected Division in MATLAB®

While we were trying to process early results, a problem arose. The difficulty was discovered to be a difference in how division by zeros was handled in different system modules. Division by zero in MATLAB® returns a value of INF, while GP returns a 1. MATLAB® does not perform protected division as a matter of course. However, it is possible to overload MATLAB® operators. After some investigation and experimentation, it was found that it is possible to overload the mrdivide (/) function of MATLAB®.

To overload a MATLAB® operator, a directory must be created, based on the type of variable upon which the operator is to work. In this case, the default data type, double, is to be overloaded. A directory labeled "@double" is used, where the overloaded function is stored. By using the addpath command to tell MATLAB® in which directory the @double directory resides, any divide operation with a value of type double will execute the new mrdivide function instead of the default MATLAB® operator.

The overloaded function was quite simple (see fig. 18). The function checks the value of the divisor; if it is zero, it returns a one; otherwise it calls the built-in MATLAB® divide operation and does double division as normal.

4.10 Processing Full Images

After the chip images are processed, the chips can be displayed with MATLAB® and evaluated. The better performing equations are used in a MATLAB® script (which is nearly identical to the script used to process the chips) to process the entire image cube through each of the selected equations. The run time depends on the number of equations used and on the complexity and length of the equations.

Figure 18. User-defined mrdivide.m function for doing protected division in MATLAB®.

```
function[x]=mrdivide(a,b)
if b==0
   x=1;
else
   x=builtin('mrdivide',a,b);
end
```

5. System Testing

5.1 Lil-gp Multiple Variable Evaluation

During the system development, several evaluation data sets were generated to evaluate each module's performance as it was developed and to evaluate the entire system.

To evaluate the modifications made in lil-gp for multiband processing, a simple evaluation file of two classes of two bands was generated (see table 5).

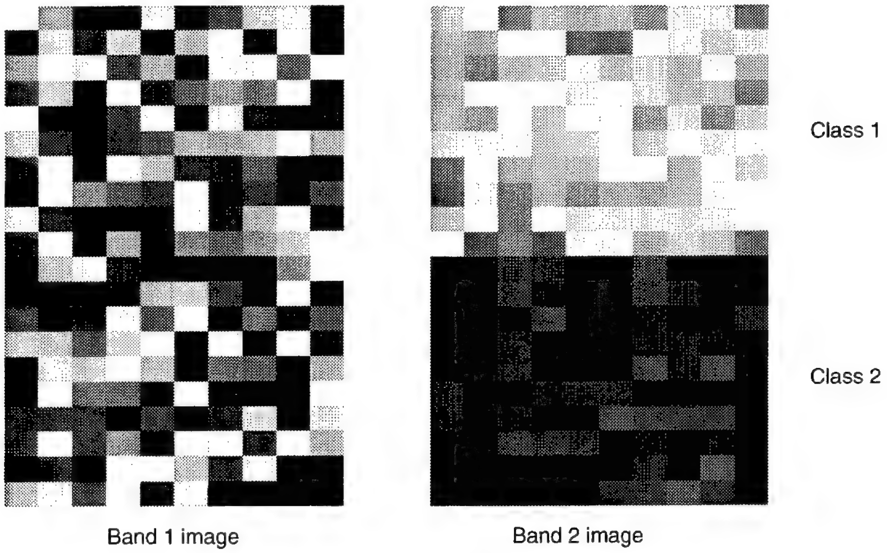
The obvious solution is to examine Band 2; if it is negative, it is class 1, if positive, it is class –1.

5.2 System Evaluation

A similar evaluation image was generated to evaluate the entire data flow. A 20-pixel-high by 10-pixel-wide image of two bands was constructed with two classes. The first band for both classes contains random values from 0 to 1. Band 2 for Class 1 also contains random values from 0 to 1, while for Class 2, it contains random values from –1 to 0. See figure 19 for an image representation of the two bands.

Table 5. Data file used for GP software development and evaluation.

Class Value	Band 1	Band 2
1	.75	–.75
–1	.3	.3
1	.6	–.6
–1	.4	.5



Band 1 image

Band 2 image

Figure 19. System evaluation image.

From this image, 10 pixels of each class were extracted by the pixel-picking tool. The two class variables were then sent to a text file with the training vector generation tool. Table 6 shows those values.

This file was supplied to the GP learning algorithms. The results of that learning are shown in figure 20.

Although it is not the most efficient representation, GP has discovered the most obvious answer (simply examine Band 2). The reason the most efficient representation is not discovered is because of the minimum tree depth constraint, which is set at two. This shows a strength of the GP method. Being unable to find the simplest solution, GP found an equivalent using +b1 and –b1 to pad the tree. Also of interest is the fact that this equation was generated in the initial population, Generation 0. For completeness, figure 21 shows the result from parsing this file. The equation on Line 2 of figure 21, after the :=, is copied and pasted into the GPimage.m program to process the whole image. The result, as expected, is the Band 2 image of figure 19.

5.3 Real Data: 28 Bands (one polarization)

The first evaluation with real data was conducted with only the 28 spectral bands of one polarization. This was done because it made sense to slowly

Table 6. Training values from evaluation image (fig. 19) presented to GP to validate modifications for multiple variables.

Number of classes	Total number of training vectors	Size of each vector
2	20	2
Class	Band 1	Band 2
1	0.606843	0.314217
1	0.485982	0.365078
1	0.485982	0.365078
1	0.891299	0.39324
1	0.198722	0.346112
1	0.198722	0.346112
1	0.418649	0.457354
1	0.465994	0.46077
1	0.746786	0.855976
1	0.0152739	0.422452
2	0.285939	–0.895202
2	0.394128	–0.942387
2	0.261819	–0.868635
2	0.783859	–0.241172
2	0.783859	–0.241172
2	0.56919	–0.85293
2	0.56919	–0.85293
2	0.014233	–0.569481
2	0.228039	–0.152594
2	0.713354	–0.791832

Figure 20. Best-of-run equation using values in figure 19 as input.

```
=== BEST-OF-RUN ===
              generation: 0
                   nodes: 5
                   depth: 2
                    hits: 20
TOP INDIVIDUAL:
-- #1 --
                    hits: 20
             raw fitness: 20.0000
    standardized fitness: 0.0000
        adjusted fitness: 1.0000
TREE:
 (- (+ b1 b2) b1)
```

Figure 21. Parser results for the previous equation.

```
OutputForm[Random #1]
f[X_]:=(((b1)+(b2))-(b1))
Expand[f[X]]
Simplify[f[X]]
Factor[f[X]]
PlotAll[f]
```

increment the data file size, since it was unknown how much time runs with real data would take. The GP code was set for 28 bands and a run was executed. Only one run with a single random seed was performed. The program successfully reached completion, resulting in a best of run equation shown in figure 22. Parsing this file resulted in figure 23.

Using this equation in the GPimage.m file and running it results in the image in figure 24. The classification chosen here is grass versus everything else because it was felt that this separation would be the easiest classification. This is believed to be a simple problem because MS data can easily produce such results with equations as simple as the NDVI. The classification that the GP is trying to make is "grass responds with values >0, everything else, <0". It is the same evaluation function used in the two-class evaluation in section 4.2. No attempt was made to find a good evaluation function for these data; yet this simple equation does appear to work somewhat. This is a very encouraging first run, especially considering that only 10 pixels of each class were presented for learning.

Compare figure 24 with figure 10. It shows that the GP solution does highlight a number of distinct objects from the grass. The pad area in the center of the image shows very nicely the narrow strip of grass that separates this area horizontally in the center. In addition, the two fence posts in the background show very well.

5.4 112 Bands

With the success of the single run for the 28 band data, runs with the full 112 bands began. Several different fitness functions were experimented with in some of the early runs.

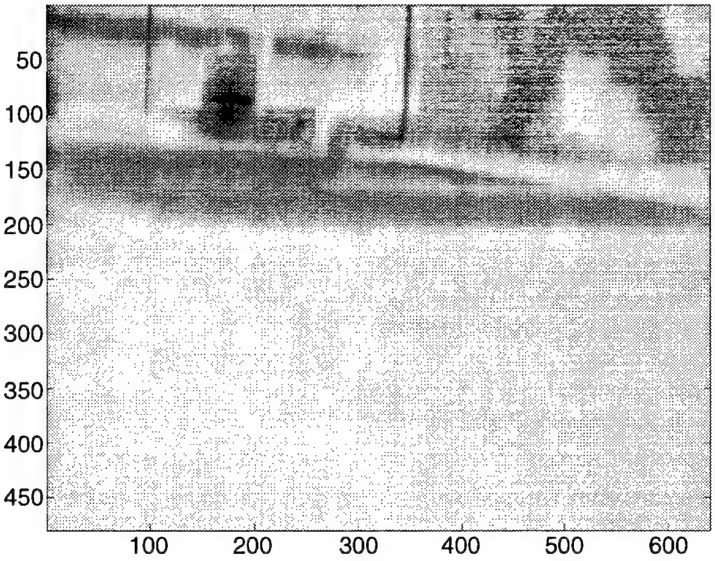
Figure 22. Best-of-run results on single polarization image.

```
=== BEST-OF-RUN ===
              generation: 14
                   nodes: 5
                   depth: 2
                    hits: 47
TOP INDIVIDUAL:
-- #1 --
                    hits: 47
             raw fitness: 47.0000
    standardized fitness: 3.0000
        adjusted fitness: 0.2500
TREE:
 (- (- b27 b8) b9)
```

Figure 23. Parsed GP equation for single polarization AOTF image.

```
OutputForm[Random #1]
f[X_]:=(((b27)-(b8))-(b9))
Expand[f[X]]
Simplify[f[X]]
Factor[f[X]]
PlotAll[f]
```

Figure 24. Classification image for grass using equation from single polarization GP run.



5.4.1 Fitness Function: Thresholding; Target > 0, Non-Target < 0

The first experiments were conducted with the same threshold case as with the 28 band evaluation in section 5.3. If a training pixel was defined as target pixel and the equation response for the pixel was greater than 0, the hit counter was incremented (a correct identification). If the pixel was non-target and the equation response was less than 0, the hit counter was incremented (also a correct identification). The total number of correct hits is the score for the equation.

This fitness condition resulted in equations with an unacceptably high false alarm rate. However, only three different random seeds were used, so it is impossible to accurately state that this fitness function is not useful, only that it has not been fully investigated.

5.4.2 Fitness Function: Thresholding; Target > 1, Non-Target < –1

To evaluate how forcing the output values farther apart would affect the performance the fitness case was modified so that if a training pixel was defined as target pixel and the equation response for the pixel was greater than 1, the hit counter was incremented (a correct identification). If the pixel was non-target and the equation response was less than –1, the hit counter was incremented (also a correct identification). The total number of correct hits is the score for the equation. This resulted in somewhat better performance of the resulting equation, but some extremely large values (± 10e34) were still found in the fully processed image.

5.4.3 Fitness Function: Bracketed; 1<Target<5, –5<Non-Target<–1

To try to reduce the number of these high values, the fitness functions were again modified to force correct values into a bound range by specifying both acceptable high and low values. If the pixel is a target pixel and the equation response is between 1 and 5 (inclusive), count this as a hit, or if the pixel is a non-target pixel and the response is between –1 and –5, count this as a hit.

Specifying lower and upper bound did improve performance of the equations evaluated. This is the fitness function used for the results discussed here. Further narrowing of the acceptable range may prove useful and should be examined in the future.

5.4.4 Fitness Function Evaluation

Most of the early runs were done only on the *grass* classification task. As mentioned earlier, it is believed that a grass (green vegetation) separation should definitely be possible because of the success of performing this separation with MS imagery such as the LandSat TM imagery. Additionally, only one or two runs were done, except for the bracketed case. Therefore, the conclusion that the bracketed case is better than the other two cases is not truly shown. However, the runs used the same starting random seeds, which should mean the starting equation population for the three fitness cases should have been identical. Thus, any performance improvements could reasonably be assumed to come from the differences in the fitness functions.

5.5 Addition of Constants as Terminals

Forcing the system to use only band values to do class separations seemed inefficient. This approach would probably tend to result in band differencing and ratioing, which may not be the best approach to the problem. Earlier work with neural networks showed that weighted sums would also work well. With only the spectral bands as terminals, however, it would be nearly impossible for GP to develop weighted sums. Furthermore, neural nets depend on their ability to use a nonlinear threshold function. The only nonlinearity that GP could use is protected division, but again, it is nearly impossible for GP to find a consistent division by zero when only using measured data as the terminals.

A solution to both problems is to add constant values to the set of terminals. A simple implementation was used to evaluate this solution, which should allow GP to at least use its nonlinearity. The terminals zero and one were added to the user-defined terminal set. To add additional terminals to lil-gp requires modification of several of the lil-gp user function files. The lil-gp instruction manual details where and how to do this (Zongker, Punch, and Rand, 1996).

Several runs were done with the bracketed fitness function. The earlier modifications of the parser had been made with the goal of making the parser as general as possible. As a result, no modifications of the parser were needed to parse the resulting equations. These results were compared to the results in section 5.4.

Again, this is not an extensive investigation—just a quick comparison. The comparison appears to show that inclusion of the terminals zero and one do improve performance. However, this conclusion may be invalid. An evaluation of batch runs for the five classes with random seeds from 1 to 23 was done with the chip images. Some batches were terminated early because of the difficulty of running larger remote jobs, and those numbers are reflected in table 7. Each parsed equation was checked for occurrences of the terminals zero and one. Table 7 shows the the use of terminals over all the classes. From the table, we see that there is a nearly even split between use of the terminal one and the use of neither of the new terminals.

However, of the 107 equations for all classes, only 30 show reasonable performance when the chip results were examined. These 30 equations were used to process the complete image cubes. Although none of the *wall* class equations perform very well, two were extracted as examples for inclusion in appendices A and B. Table 8 examines the use of the constant terminals in the acceptable equations. In general, a similar split can be seen (ignoring the wall equations since not even the two listed perform adequately).

The addition of other terminal values such as 0.5, 0.25, or perhaps a series of evenly spaced values from zero to one (10 or 100 samples) may further improve performance. This may allow the evolution of weighted band values. Weighted band values may allow GP to develop more general equations. These equations may appear more like PCA or neural network solutions since weighted bands are the basis of these techniques.

Table 7. Number of equations that used the new terminals.

Terminals	All classes Total=107	Grass Total=23	Pad Total=22	Wall Total=21	Barrel-Dark Total=21	Barrel-Bright Total=20
Only 1	50	12	10	8	4	16
Only 0	10	1	3	3	2	1
Both	4	0	2	1	1	0
Neither	43	10	7	9	14	3

Table 8. Number of "useful" equations that used the new terminals.

Terminals	All classes Total=32	Grass Total=7	Pad Total=5	Wall Total=2	Barrel-Dark Total=9	Barrel-Bright Total=9
Only 1	13	2	2	0	1	8
Only 0	4	1	1	1	1	0
Both	1	0	0	0	1	0
None	14	4	2	1	6	1

6. Evaluation Image Results

Based on the evaluation of the image chips, the 32 "best" equations were used to process the training image and the three evaluation images. Observing the processed training image, it is difficult to say which if any equation for a particular classification performs better than the others. However, by examining the equations' performance on the three evaluation images, it is possible to evaluate how general an equation is. Examining each equation's performance over the four images, the two "best" performing equations for each classification were selected. Appendix B contains the four classification images for each of the equations selected.

6.1 Processing the Entire Training Image

Figures 25 and 27 show two of the more favorable results on the training image. It is impressive that GP, with the simple fitness function and simple operators used, performs as well as it does. The added advantage of the GP results is access to the actual equations that produce these classification images (figs. 26 and 28 show the equation for the image shown in figs. 25 and 27 respectively).

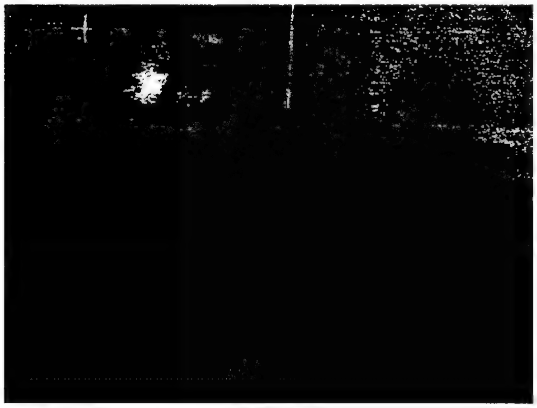
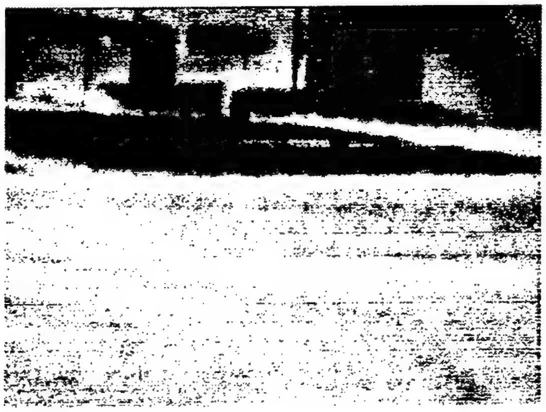


Figure 25. Classification images for barrel bright; GP equation 5.

```
((((((((((((b23)-(b52))+((((b101)*(b111))*(b33))*(b56)))-((b26)+(b3)))+(((((((b27)-(b55))+(((b70)/(b5))
+((b43)*(b112))))+((b83)(b72)))+((b97)*((((b97)+(b13))+((b107)+ (b20)))+(b112))))+ (((b101)*(b111))
*((b67)-(b81))))+((b97)*(((b55)+((b107)+(b20)))+ (b112)))))-((b36)*(b34)))*(((b96)-(b37))/((b28)
+(b34))))*(b33))-((b32)+(b75)))-((1.000000)+(((b7)+((b97)+(b23)))/((((b8)+(b112))+ (((b100)*(((b101)
*(b111))*(((b70)+ ((b105)+(b19)))-(b33))))/((((b67)-(b81))+((b10)/(b59)))* (((b83)/(b4))/((b32)/(b19))))))
-(((b96)-(b37))*(b32))))))-(b107))-(((b7)+((b97)+(b23)))/((((b8)+ (b112))+(((b100)* (((b107)+(b20))
+(((b68)-(b104))*(((b1)-(b5))+(((b37)*(b43))*((b100)+(b84)))))))/ ((((b67)-(b81))+((b10)/(b59)))*(((b83)
/(b4))/((b32)/(b19))))))-(b83))))
```

Figure 26. Parsed GP equation 5 used to generate figure 25.

Figure 27. Classification images for Grass; GP equation 1.



```
((((b44)+(b8))+(((((b88)+(b27))+(((b101)+(b17))+((b13)-(b97))))+(b112))-((b28)*(((((b26)+(b36))+((b17)
+(b75)))*((b2)+(((b97)+(((b51)*(b87))-((b25)*(b16))))+ ((b62)+(b21)))))*(((b90)+(b14))*((((b62)+(b21))
+((b41)-(b86)))-(((b30)+(b81))-((b111)-(b63)))))))))+(((((b86)/(b70))*(((b29)-(b49))*((b11)-(b109))))
/(((b108)+(b22))+ (b28)))*((b29)-(b63))))
```

Figure 28. Parsed GP equation used to generate figure 27.

6.2 Processing Evaluation Images

While the results look promising, there is one concern. This image is the cube from which the training data were extracted. While it is significant that these results can be obtained with only 10 training pixels of each class, how will these equations perform on other cubes? The three other cubes extracted from the data set available are used to examine how general a solution these equations might be. Each of these cubes is very different. The first is labeled *Fence* (see fig. 29). It basically is a field with a brush line in the background and some fence posts in the foreground. The second is labeled *Trucks* (see fig. 30). It appears to be an image of a storage/parking area, containing several vehicles, with a fence in the background. The final cube is labeled *Van* (see fig. 31). It contains a passenger van in a grassy field. A target board of some sort is on the left edge of the image. In some of the result images a pickup truck appears to be in the background. Appendix B contains the processed classification images from these cubes.

However, without accurate ground truth for these images, it is impossible to accurately determine which equations are correct and which are not. For example, look at the *Trucks* image processed through the *barrel-dark* equations Numbers 8 and 9 (see fig. 32).

These are very different results. However, without knowledge of the materials (or more precisely their spectra) in the images, it is impossible to say with any confidence which equation processes the image cube better.

What about the two classification images in figure 33 processed for *barrel bright*?

Without spectral sampling of the paint on the trucks, it cannot be determined if the paint is similar to the bright paint on the barrel. If it is similar, then equation 5 is the better equation; if not, equation 1 is.

Figure 29. Evaluation image *Fence*.

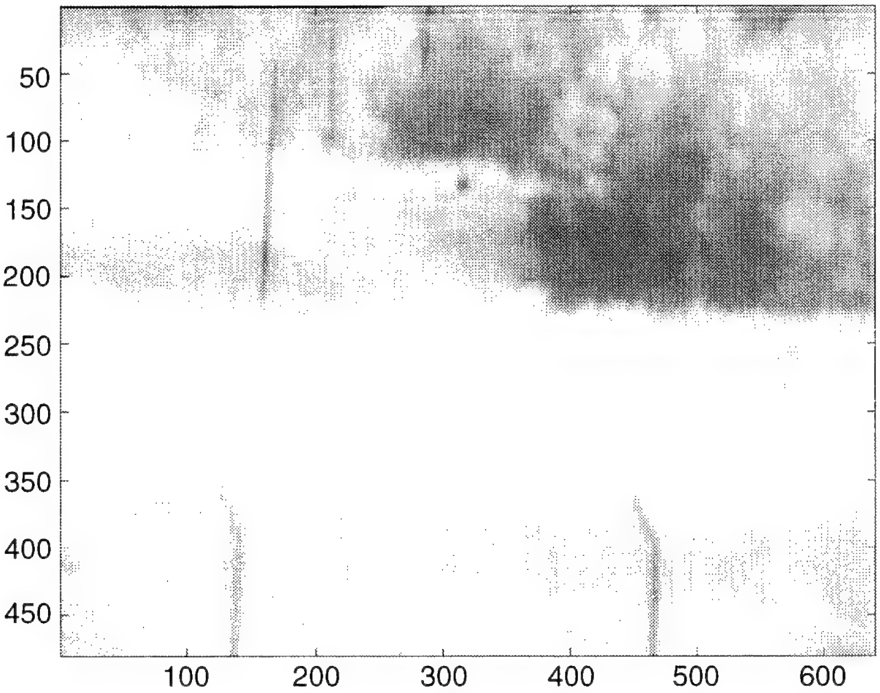


Figure 30. Evaluation image *Trucks*.

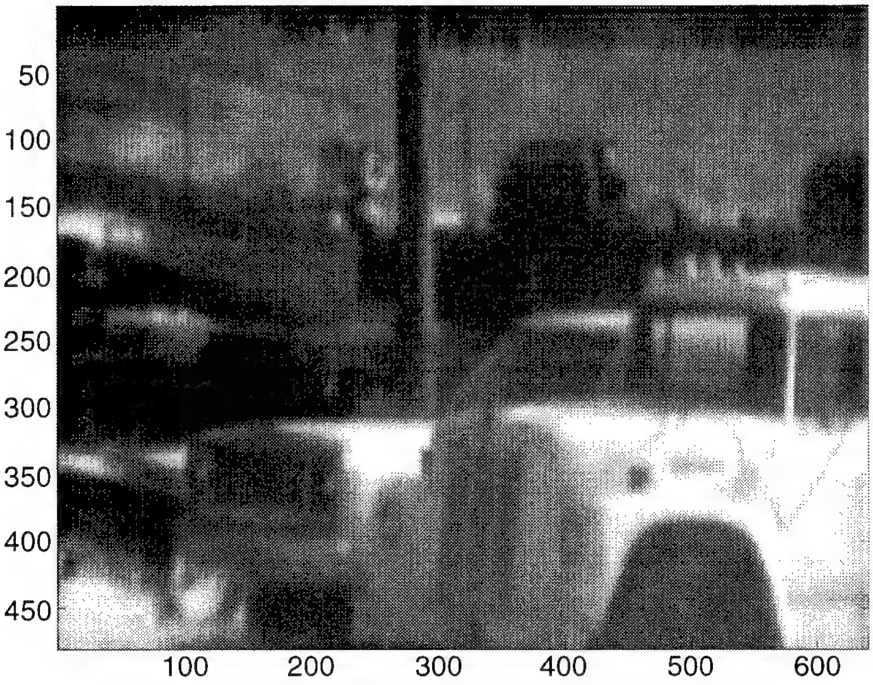


Figure 31. Evaluation image *Van*.

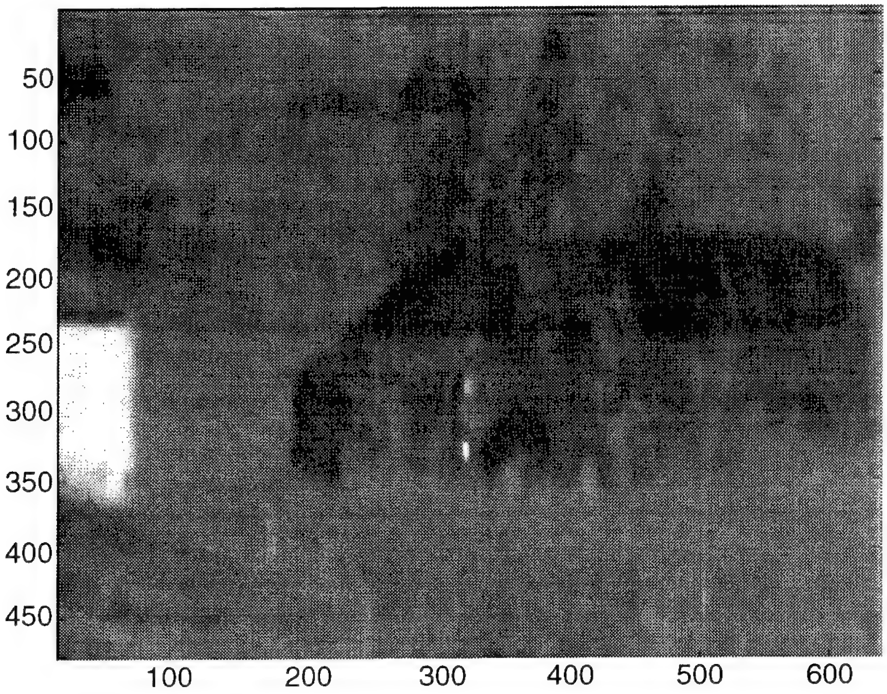


Figure 32. Evaluation image *Trucks* processed with *barrel-dark* equations. (A: equation 8, B: equation 9.)

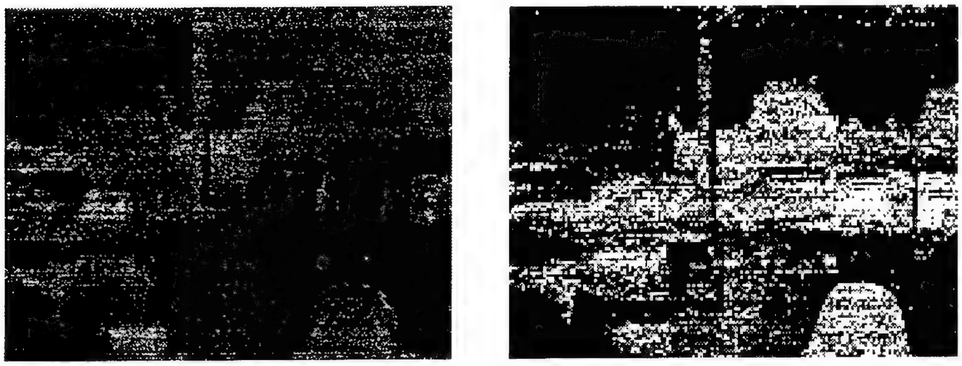
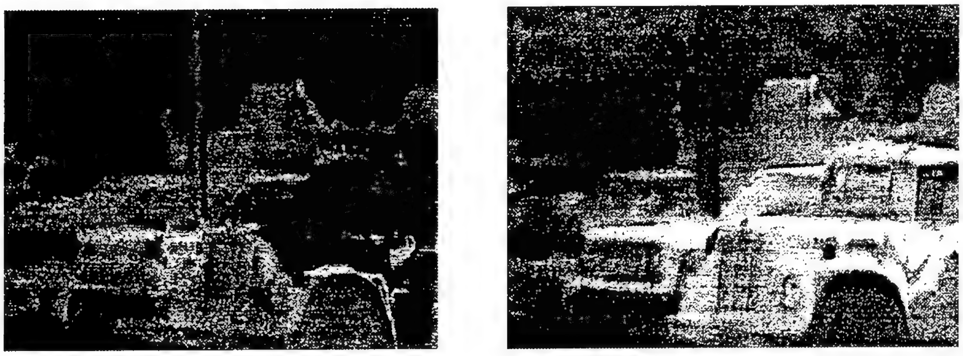


Figure 33. Evaluation image *Trucks* processed with *barrel-bright* equations. (A: equation 1, B: equation 5.)



7. Conclusions

This report has documented the first known application of genetic programming techniques to the processing of HS and SP imagery. The results show that this technique has significant potential for use in data mining and data stream compression as well as in target detection and material classification. The infrastructure is now in place to continue development of this technique.

7.1 Areas for Future Investigation

Clearly, many areas remain for future investigation. Some of the more obvious are

- How would more training samples affect performance?
- Would training samples from more than one image cube improve performance?
- Does the addition of the terminals 0 and 1 really make a difference?
    - What about adding terminals between 0 & 1?
    - What sampling over the 0-to-1 interval (how many) to use?
    - Should the sampling be uniform or random?
- How does the fitness function affect GP performance?
    - Is the bracketed range truly better than either of the unbound thresholds?
    - How would a fitness function forcing responses to equal –1 and 1 perform?
    - How would more sophisticated fitness functions perform and what would they look like?
- How general are the solutions?
- How do these solutions compare to other techniques?
- How would GP perform on different data sets such as AVIRIS, SEBASS, or FTHSI?

No attempt has yet been made to analyze the current set of equations as to what bands are being used. These may prove to be useful bands for the compression of spectral data for transmission on the battlefield. Additionally, looking at how the bands, constants, and operators are being used may give some insight into the process for HS and SP detection and material recognition.

Acknowledgments

This work was conducted primarily from September 1999 to April 2000, in the Atmospheric, Oceanic, and Space Sciences Department of the School of Engineering at the University of Michigan, Ann Arbor, as a Master's research project under the guidance of Prof. Jason Daida. I would like to thank Prof. Daida for his guidance and insights in this project. I would like to also thank Shahbaz Chaudhary for his invaluable work on modifying the lil-gp code for multiple variables.

The evaluation images were processed while I was on a Federated Laboratory rotation supporting the Advanced Sensors Consortium at Veridian ERIM International, Ann Arbor, MI.

I also thank Dr. Neelam Gupta of ARL for the AOTF data used for this project.

References

Adler-Golden, S., Anderson, G. P., Pukall, B., Allred, C. L., Jeong, L. S., Hoke, M., Chetwynd, J., Berk, A., Bernstein, L. S., Richtsmeier, S. C., Acharya, P. K., and Ma, M. W. (1999). FLAASH and MODTRAN4: State of the Art Atmospheric Correction for HyperSpectral Data, *Proc. 1999 IEEE Aerospace Conf.*, Vol. 4, pp 177–181.

Banzhaf, W., Nordin, P., Keller, R. E., and Francone, F. D. (1998). *Genetic Programming—An Introduction*, Morgan Kaufman Publishers, San Francisco, CA.

Brumby, S. P., Theiler, J., Perkins, S. J., Harvey, N. R., Szymanski, J. J., Bloch J. J., and Mitchell, M. (1999). Investigation of Image Feature Extraction by a Genetic Algorithm, In *Proceedings of SPIE 3812*, pp 24–31.

Clark, R. N. and Swayze, G. A. (1995). Mapping Minerals, Amorphous Materials, Environmental Materials, Vegetation, Water, Ice and Snow, and Other Materials: The USGS Tricorder Algorithm, In *Summaries of the Fifth Annual JPL Airborne Earth Science Workshop*, R.O. Green, Ed., JPL Publication 95–1, pp 39–40.

Cleva, C., Cachet, C., Cabrol-Bass, D., and Forrest, T. P. (1997). Advantages of a Hierarchical System of Neural-networks for the Interpretation of Infrared Spectra in Structure Determination, *Analytica Chimica Acta*, 348, pp 255–265.

Cowe, I. A. and McNicol, J. W. (1985). The Use of Principal Components in the Analysis of Near-Infrared Spectra, *Applied Spectroscopy*, Vol. 39, No. 2, pp 257–266.

Crist, E. P. and Cicone R. C. (1984). Application of the Tasseled Cap Concept to Simulated Thematic Mapper Data, *Photogrammetric Engineering and Remote Sensing*, Vol. 50, No. 3, pp 343–352.

Daida, J, M., Hommes, J. D., Bersano-Begey, T. F., Ross, S. J., Vesecky, J. F. (1996). Algorithm Discovery Using the Genetic Programming Paradigm: Extracting Low-Contrast curvilinear features from SAR Images of Arctic Ice, In P. Angeline and K. Kinnear Jr. (ed.) *Advances in Genetic Programming II.* Cambridge, MA: The MIT Press, pp 417–442.

Daida, J. M., Onstott, R. G., Bersano-Begey, T. F., Ross, S. J., and Vesecky, J. F. (1996). Ice Roughness Classification and ERS SAR Imagery of Arctic Sea Ice: Evaluation of Feature-Extraction Algorithm by Genetic Programming. In *Proceedings of the 1996 IGARSS*, Washington: IEEE Press, pp 1520–1522.

Daida, J. M., Bertram, R. B., Polito, J. A., and Stanhope, S. A. (1999). Analysis of Single-Node (Building) Blocks in Genetic Programming. In L. Spector, W.B. Langdon, U.M O'Reilly, and P.J. Angeline (Eds.), *Advances in Genetic Programming 3.* Cambridge: The MIT Press, pp 217–241.

Daida, J. M., Polito 2, J. A., Stanhope, S. A., Bertram, R. R., Khoo, J. C., Chaudhary, S. A., and Chaudhri, O. (2000). What Makes a Problem GP-Hard? Analysis of a Tunably Difficult Problem in Genetic Programming, In *Journal of Genetic Programming and Evolvable Hardware*. Accepted

Denes, L. J., Gottlieb, M., Kaminsky, B., and Huber, D. F. (1997). Spectro-Polarimetric Imaging for Object Recognition, *SPIE Proceeding 3240*, Paper #02.

Der, S. and Chellappa, R. (1997). Probe Based Automatic Target Recognition in Infrared Imagery, *IEEE Trans. On Image Processing*, Vol. 8, No. 1 pp 92–101.

Drake, K. C., Kim, R. Y., and Kim, T. C. (1993). Pixel-level Object Segmentation from Multispectral Sensor Imagery, Proc. *SPIE 22nd Applied Imagery Pattern Recognition Workshop*, Washington, D.C., Oct. 1993.

Goward, S. N., Markham, B., Dye, D. G., Dulaney, W., and Yang, J. (1991). Normalized Difference Vegetation Index Measurements from the Advanced Very High Resolution Radiometer, *Remote Sensing of the Environment*, Vol. 35, pp 257–277.

Guenther, R. (1990). *Modern Optics*, John Wiley & Sons, New York, NY, Chapters 2 & 3.

Gupta, N., Dahmani, R., Gottlieb, M., Denes, L., Kaminsky, B., and Metes, P. (1999). Hyperspectral Imaging using Acousto-Optic Tunable Filters, *SPIE Proceedings 3718*, pp 512–521.

Harsanyi, J. C. and Chang, C. (1994). Hyperspectral Image Classification and Dimensionality Reduction: An Orthogonal Subspace Projection Approach, *IEEE Trans. Geosci. and Remote Sensing*, Vol. 32, No. 4, pp 779-785, July 1994.

Hecht-Nielsen, R. and Zhou, Y. (1995). VARTAC: A Foveal Active Vision ATR System, *Neural Networks*, Vol. 8, Issue 7-8, pp 1309–1321.

Howard, D. and Roberts, S. (1999). A Staged Genetic Programming Strategy for Image Analysis, In GECCO-99: *Proceedings of the Genetic and Evolutionary Computation Conference*, W. Banzhaf, et al. (Eds.) Morgan Kaufmann, San Francisco: pp 1047–1052.

Jiaju, L. (1988). Development of Principal Component Analysis Applied to Multitemporal Landsat TM Data, *International Journal of Remote Sensing*, Vol. 9, No. 12, pp 1895–1907.

Kaufman, Y. J. and Tanre, D. (1992). Atmospherically Resistant Vegetation Index (ARVI) for EOS-MODIS, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 30, No. 2, pp 261–270.

Kauth, R. and Thomas, G. (1976). The Tasseled Cap–A Graphic Description of Spectral-Temporal Development of Agricultural Crops as Seen by Landsat, Proceedings *2nd International Symposium on Machine Processing of Remotely Sensed Data*, Purdue Univ.

Koza, J. R. (1992). *Genetic Programming: On the Programming of computer by Means of Natural Selection,* Cambridge, MA: MIT Press.

Koza, J. R., Bennett III, F. H., Adre, D., and Keane, M. A. (1999). *Genetic Programming III: Darwinian Invention and Problem Solving,* Morgan Kaufmann Publisher, San Francisco, CA.

Lee, J. B., Woodyatt, A. S., and Berman, M. (1990). Enhancement of High Spectral Resolution Remote Sensing Data by a Noise-adjusted Principal Components Transform, *IEEE Trans. Geosci. Remote Sensing,* Vol. 28, pp. 295-304, May 1990.

McKee, D. and Bandera, C. (1998). Multistage Foveal Target Detection System, SPIE Aerosense '98.

Rauss, P. J. (1992). Automatic Recognition of Spectral Components using Neural Networks, Proc. *Second ATR Systems and Technology Conference,* GACIAC, ITT Research Institute, Vol. 2, pp 169–188.

Rauss, P. J., Cederquist, J., Dwan, C., and Wegrzyn, J. (1999). Spatial Spectral Target Detection*, Proceedings of 7th ATR Systems and Technology Symposium,* Monterey CA,

Rauss, P. J., Daida, J. M. and Chaudhary, S. (2000). Classification of Spectral Imagery using Genetic Programming. *GECCO-2000: Proceedings of the Genetic and Evolutionary Computation Conference, July 10–12, 2000, Las Vegas, Nevada.* San Francisco: Morgan Kaufmann Publishers.

Richter, R. (1996). Atmospheric Correction of DAIS Hyperspectral Image Data, *Computers & Geosciences,* Vol. 22, No. 7 pp. 785–793.

Roberts, S. and Howard, D. (1999). Evolution of Vehicle Detectors for Infrared Line Scan Imagery, *EvoIASP'99 and EuroEcTel'99 Joint Proceedings,* Springer LNCS, pp 111–125.

Slater, P. N. (1980). *Remote Sensing: Optics and Optical Systems,* Addison-Wesley Publishing Co., Reading, MA. pp 204–206 & 234, 1980.

Smith, M. O., Johnson, P. E., and Adams, J. B. (1985). Quantitative Determination of Mineral Types and Abundances from Reflectance Spectra using Principal Components Analysis, in Proc. *15th Lunar and Planetary Sci. Conf., Part 2, Geophys. Res.,* Vol. 90, Suppl., pp. C797–C804, February 15.

Tackett, W. (1993). Genetic Programming for Feature Discovery and Image Discrimination. *Proceedings of the Second International Conference on AI Planning Systems,* pp 669–673.

Wang, L. C., Der, S., and Nasrabadi, N. (1998). Automatic Target Recognition using a Modular Neural Network, *IEEE Journal of Image Processing,* Vol. 7, No. 8, pp 1113–1121.

Wolff, L. B. (1995). Applications of Polarization Camera Technology, *IEEE Expert,* October 1995 issue, pp 30–38.

Zongker, D. and Punch, W. (1995). lil-gp, Lansing, Michigan State University, GA Research and Applications Group. http://garage.cps. msu.edu/ software/ software-index.html

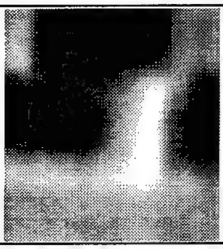
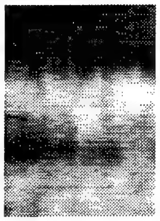
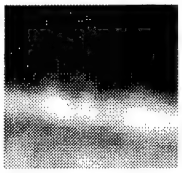
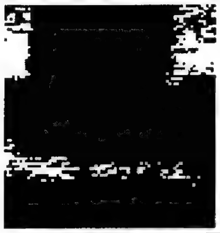
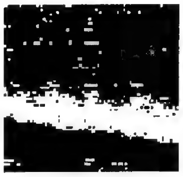
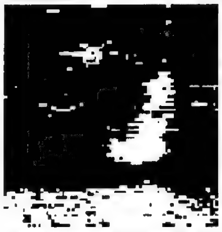
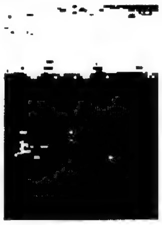
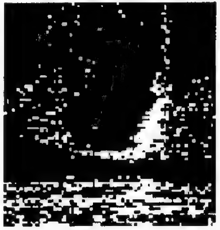
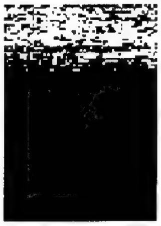
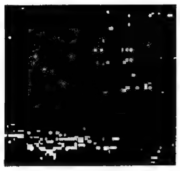
Zongker, D., Punch, W., and Rand, B. (1996). lil-gp 1.01 User's Manual, supplied with software available at http://garage.cps.msu.edu/ software/ software-index.html

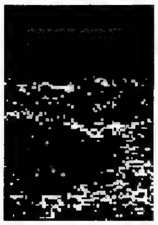
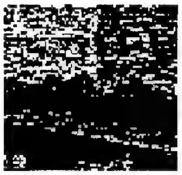
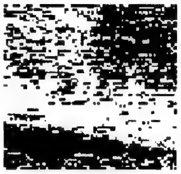
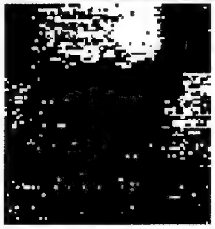
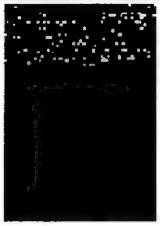
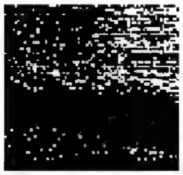
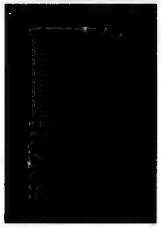
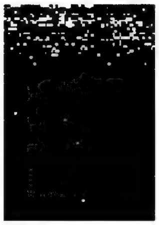
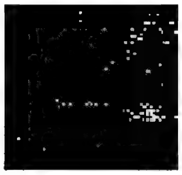
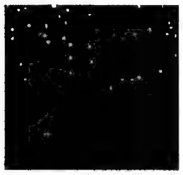
Appendix A. Best Equations

This table shows the two best equations of the 20 to 23 equations generated for each class (see sect. 4.9 for details). For each target class, all the chips were displayed and subjectively rated using a scale from 1 to 10 to rate their potential usefulness as a classification equation for the image. Each was judged on how well it performed in highlighting the desired class and how well it rejected pixels of the other classes. The desired class is displayed in white and the other classes as black.

Because of the way GP calculates its output values, there is no limit on what an equation's response to a particular pixel can be. Since these equations were generated with the bracketed fitness function (see sect. 5.4.3), it was decided to display the results as a binary image. The gray scale was set to saturate at black at a value of –0.01 and to white at 0.01, effectively performing a binary threshold at zero. This removes the stretching that can occur because of extreme values returned from some of the pixels. All the chips and full images generated have some extremely high values, both positive and negative. If the image is allowed to adjust the gray scale to these high and lows, the entire image is a uniform gray except for a very few white and black pixels that represent these outliers.

The first row of the table shows the chip labels. The second row shows a gray scale intensity image of the clearest single band in the training image cube, Band 72. The various classes can clearly be seen in these chips.

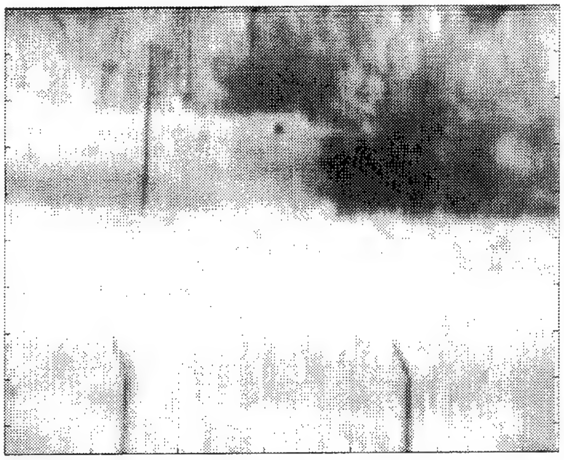
	Barrel chip	Pad chip	Wall chip
Band 72			
Grass Equation 1			
Grass Equation 13			
Pad Equation 9			
Pad Equation 22			

Wall Equation 7			
Wall Equation 12			
Barrel – Dark Equation 4			
Barrel – Dark Equation 9			
Barrel – Bright Equation 13			
Barrel – Bright Equation 14			

Appendix B. Processed Images

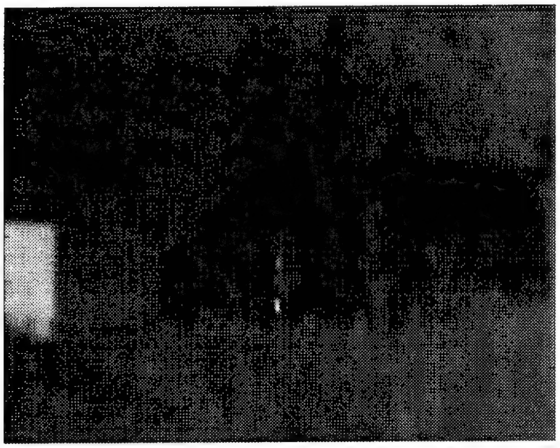
This appendix shows the classification images resulting from processing the training image and the three evaluation images through the two "best" GP equations for each class separation. The gray scale on the images was set to saturate at ±0.01, which results in a nearly binary image thresholded at zero. The images show the desired class as white, as determined by the equation's value for that pixel. The other four classes respond with negative values and are displayed in black. The equation used to process the image cube is shown on pages 59–68. Gray scale images of Band 72 for each of the images are shown on page 58.

Barrel image



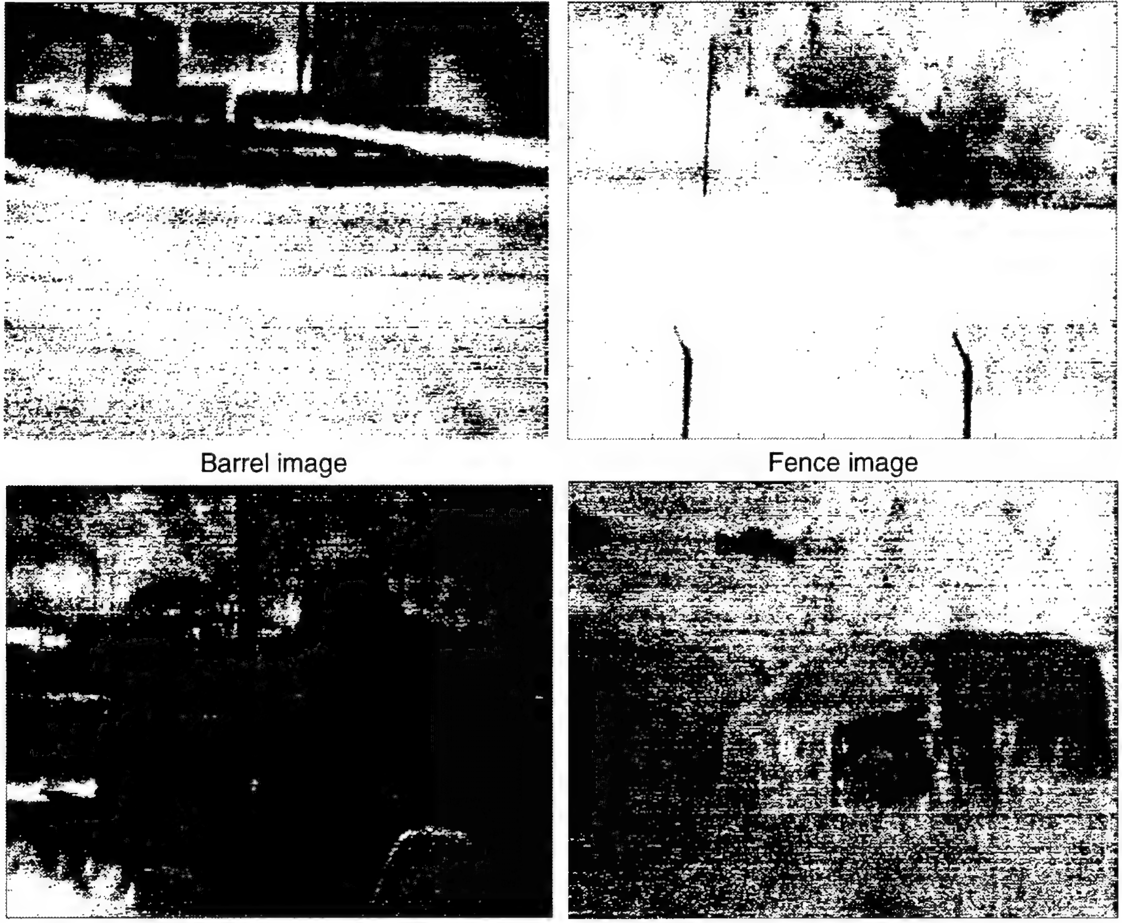
Fence image

Trucks image



Van image

Grass equation 1



Barrel image

Fence image

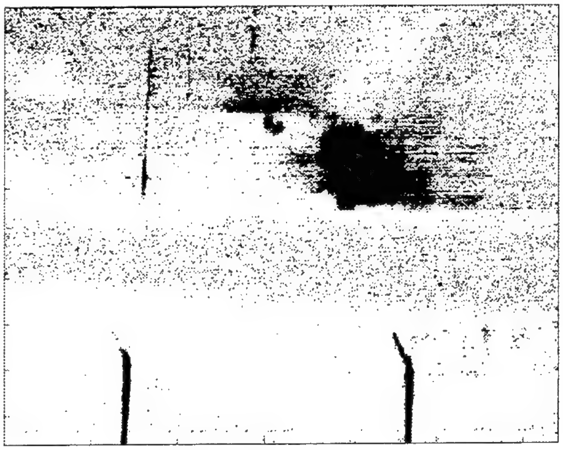
Trucks image

Van image

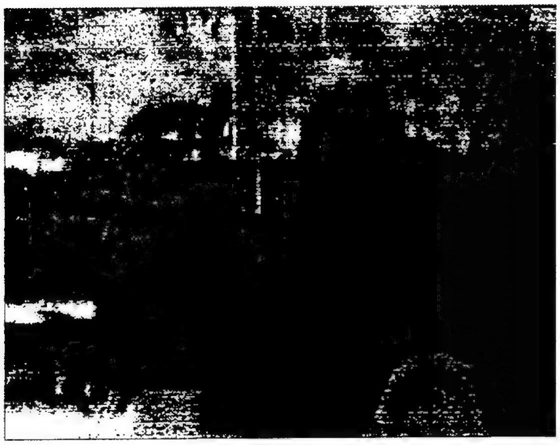
((((b44)+(b8))+(((((b88)+(b27))+(((b101)+(b17))+((b13)-(b97))))+(b112))-((b28)*(((((b26)+(b36))+((b17)
+(b75)))*((b2)+(((b97)+(((b51)*(b87))-((b25)*(b16))))+((b62)+(b21)))))*(((b90)+(b14))*((((b62)
+(b21))+((b41)-(b86)))-(((b30)+(b81))-((b111)-(b63)))))))))+(((((b86)/(b70))*(((b29)-(b49))*((b11)
-(b109))))/(((b108)+(b22))+(b28)))*((b29)-(b63))))

Grass equation13

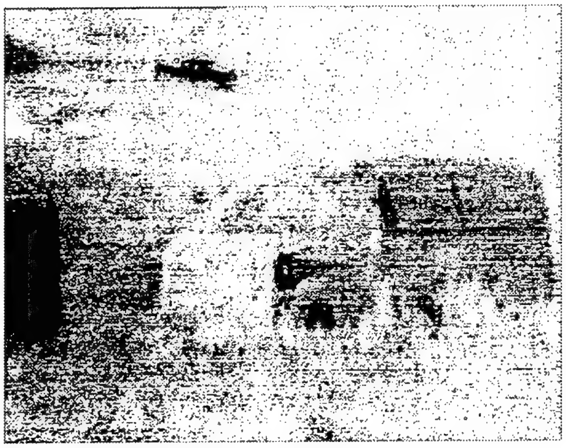
Barrel image



Fence image



Trucks image

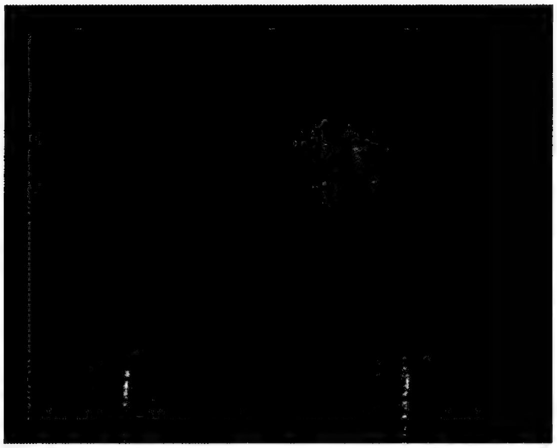


Van image

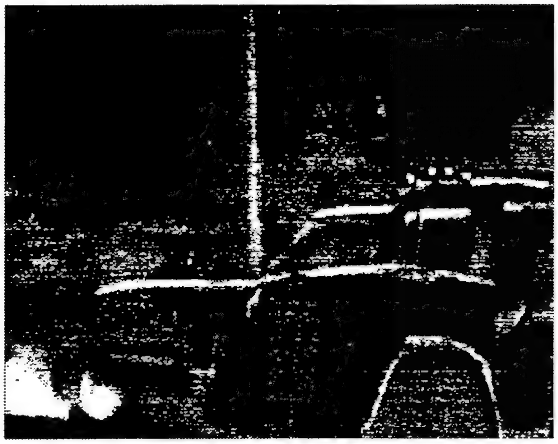
(((((b28)+(b38))-(((b25)+(b111))*((b75)+(b53))))*((((b42)/(b13))+((b106)+(b58)))-(((b60)-((b28)+(b38)))
*(b77))))-((((((b2)/(((((b28)+(b38))+((b3)-(b11)))-(((b109)-(b10))-((b32)-(b22))))+((((b60)-((b83)+(b81)))
((b55)(b76)))/(((b31)+((b45)/(b40)))/((b64)/(b105))))))-((((b101)-(b69))*((b111)*(b25)))-((((b29)
+(b30))-((b105)-(b78)))*(((b17)*(b80))*((b15)-(b81))))))*((b97)*(b43)))-(((b89)+(b8))-((b58)*(b80))))
/((((b28)+(b38))+((b3)-(b11)))-(((b109)-(b10))-((b32)-(b22))))))

Pad equation 9

Barrel image



Fence image

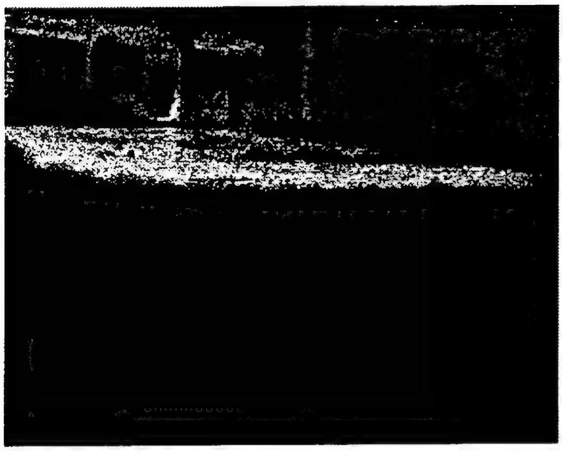


Trucks image

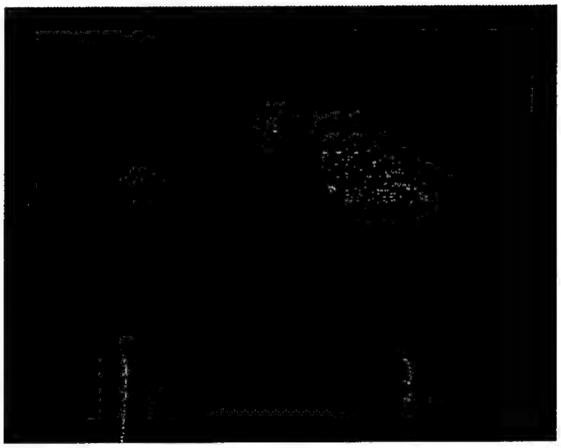
Van image

```
((((b11)+(b65))/((((b24)+(((b1)+(b111))+(((b6)*(b1))*((b107)+(b24)))))*((b59)*(b71)))+((b108)
/(b10))))-((((b24)+(((b1)+(b111))+(((b79)*(b98))*((b59)*(b71)))))*(((b59)*(b71))+((b84)*(b84))))
+((b108)/(b10))))
```

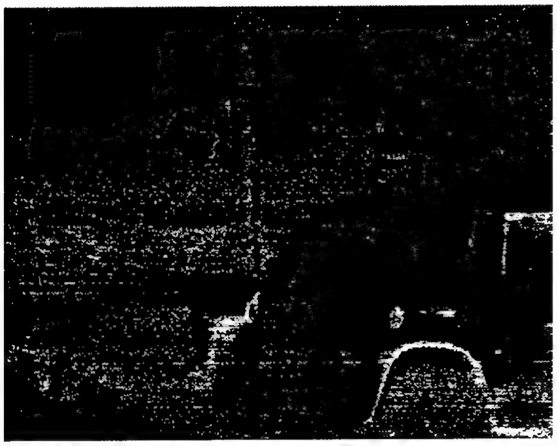
Pad equation 22



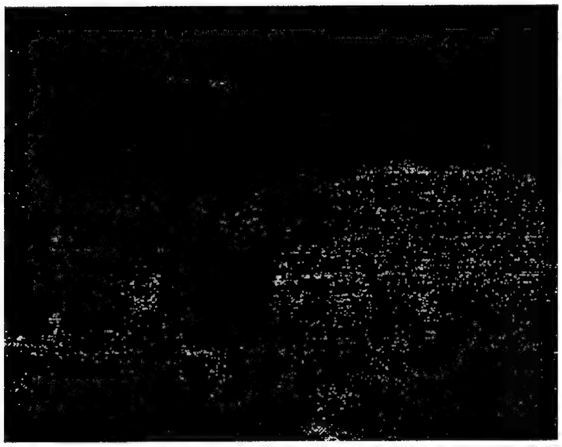
Barrel image



Fence image



Trucks image

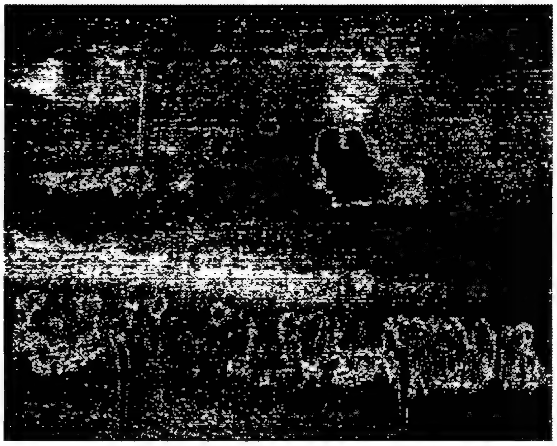


Van image

```
((b18)-(((((b4)/(b31))*((b60)-(b39)))*(((b47)+((((b110)+(b110))-(((b112)+(b90))-(((((b95)/(b5))*((b47)
+(b18)))+((b14)+(b96)))+(b71))))/(b112)))*((b21)-(b4))))-((((b6)-(1.000000))-((((((((b18)+(((b18)+(b41))
-(((b93)+(b74))-((b96)-(b44)))))+(((b66)*(b72))+((b60)-(b39))))*((b89)-(((b80)/(b15))*((b21)-(b4)))))
-((b22)-(((((b95)/(b5))*((b80)/((b6)-(1.000000))))+((b14)+(b96)))+(b71))))/(b112))/((b9)+(b16)))*((((b85)
*(b101))*((b66)*(b72)))/(((((b44)/((b37)+(b112)))-(b89))+((((b42)/(b81))-(((b44)*(b3))*((b24)+(b30))))
/((b39)/((b30)*(b94)))))/(((b16)-(b49))+((b63)*(b59)))))))-(((((b44)*(b3))*((b24)+(b30)))/(b21))
*((((b85)*(b101))*((b66)*(b72)))/(((b19)+(b2))/((1.000000)+(((b95)/(b5))*((((b4)/(b31))*(b45))*(((((b4)
/(b31))*((b60)-(b39)))*(((b47)+(b18))*((b21)-(b4))))*((b21)-(((((b4)/(b31))*((b60)-(b39)))-(b46))-(((b47)
+(b18))*(b5))))))))))))))
```

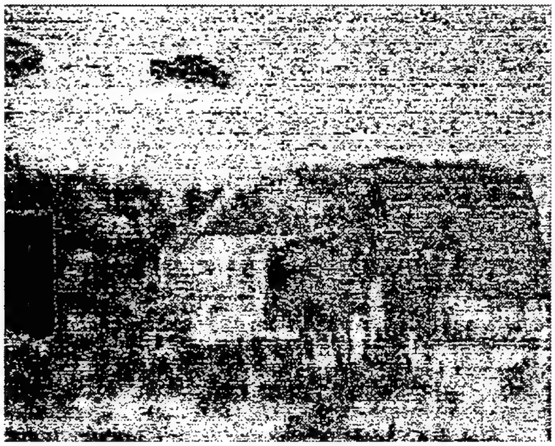
Wall equation 7

Barrel image



Fence image

Trucks image



Van image

```
((((((((b55)*((b83)*(b57)))-((b11)-((b49)+(b91))))/(((((((b102)-(b88))-((b35)+(b79)))-((b107)-(b62)))
+(b23))-((b11)/(b12)))-(((b21)/(b108))-(((((b89)+(b54))-((b61)+(b57)))-((((((b89)+(b54))-((b61)+(b57)))
-((b78)-(b100)))+((b62)/(b81)))*((b86)*(b107))))+((b62)/(b81))))))+((b13)+(b25)))+((((((b52)-(b84))
/(((((b90)*(b27))-((b90)/(b45)))+(((b13)+(b25))-((b91)*(b111))))+(((((b102)-(b88))-((b98)+(b112)))
-(((b67)-((b80)+(b43)))*(b79)))-(((b91)*(b111))-((b2)/(b14))))))-((b98)+(((b52)-(b84))/((((b24)-(b58))
+(((b13)+(b25))-((b91)*(b111))))+(((((b102)-(b88))-((b98)+(b112)))-(((b21)/(b108))*(b82)))-(((b91)
*(b111))-((b2)/(b14))))))))-(((b40)-((b13)+(b97)))*(b82)))-(((((((b67)-((b80)+(b43)))*(b79))-((b11)/(b12)))
-((b68)*(b9)))*(((b71)-(b33))+((((((b57)/(b55))/(((b11)-((b35)+(b79)))-(((b90)*(b27))-((b90)/(b45)))))
+(((b105)+(b95))-((b57)/(b55))))-(b78))-(((b92)*((((b83)*(b9))-((b49)+(b91)))-(((b102)-(b88))-((b98)
+(b112)))))-((b2)/(b14))))))-((((((b57)/(b55))-(((b83)*(b9))-((b49)+(b91))))+(b23))-((b11)/(b12)))
-((((b90)*(b27))+((b62)/(b81)))*((b86)*(b107))))+(((b105)+(b95))-((((b89)+(b54))-((b61)+(b57)))-((b78)
-(b100))))))))*(((b90)*(b27))-((b63)*(b6))))-(((b57)/(b55))-((b2)/((b57)/(b55)))));
```

Wall equation 12

Barrel image

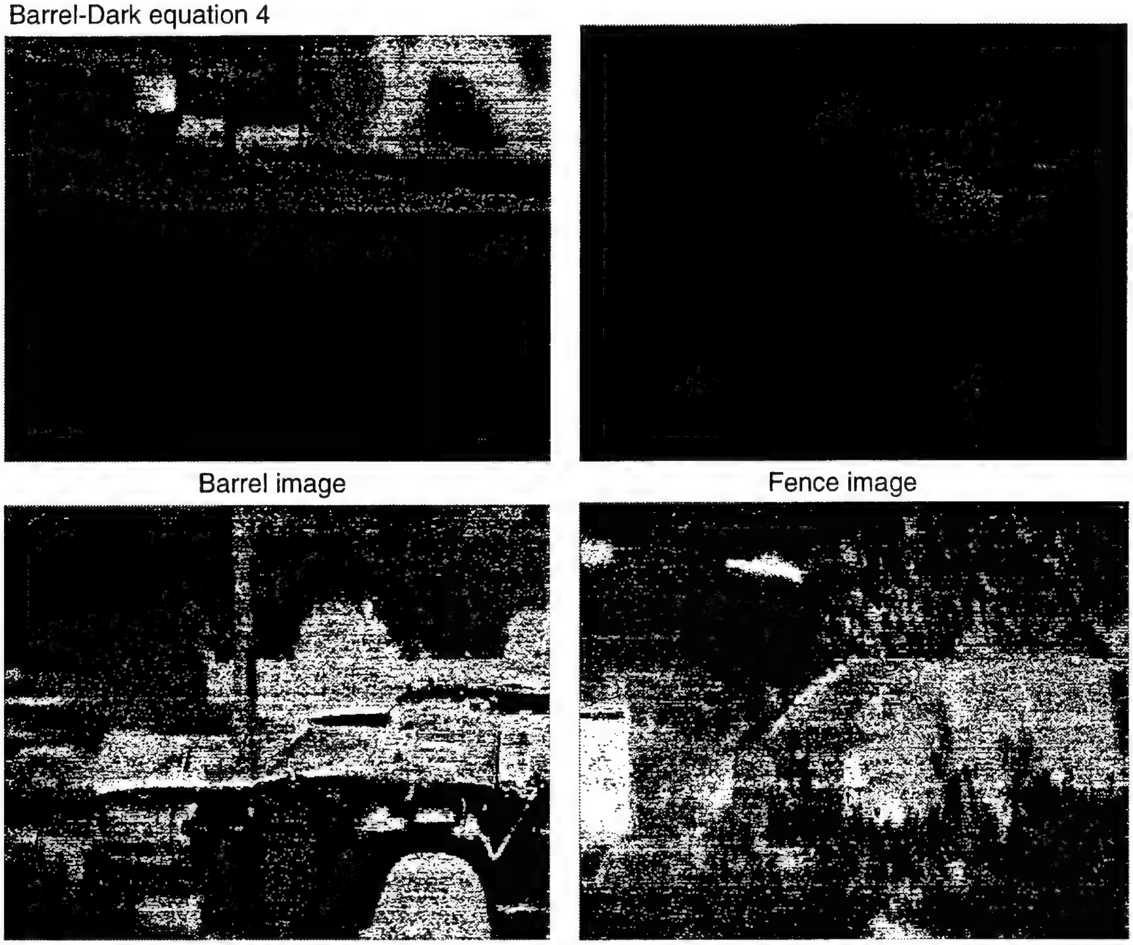
Fence image

Trucks image

Van image

((((((b55)/((((((b28)*(b108))/(((b87)-(b35))/((b2)+(b13))))/((0.000000)-((((b103)-(b73))*(b6))+((b65)
+(b62))))))/(((b19)+(b95))-((b12)+(b13))))-((((b103)-(b73))-((b31)+(b72)))*(((((((b22)+(b95))
/((b19)+(b95)))-((b24)+(b12)))+((b65)+((b87)+(b26))))/((b19)+(b95)))-((b46)/(b43))))))+(b17))*((b28)
((b46)/(b43)))))/((b103)-(b73)))+((((((b87)+(b17))-((((((b12)+(b13))-(((b103)-(b73))-((((b47)((b28)
(b108)))/(((b108)(b86))/((b19)+(b95))))/((b19)+(b95)))))-(((b2)-((b87)+(b26)))/((b50)-(b35))))
+(b8))*(b25)))+(((b2)-(((b19)+(b95))-((b57)/(b6))))/((b50)-(b35))))+((b19)+(b17)))-((b24)+(b12))))

Barrel-Dark equation 4



Barrel image

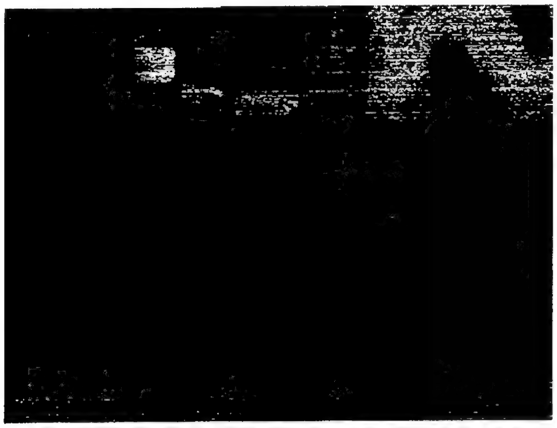
Fence image

Trucks image

Van image

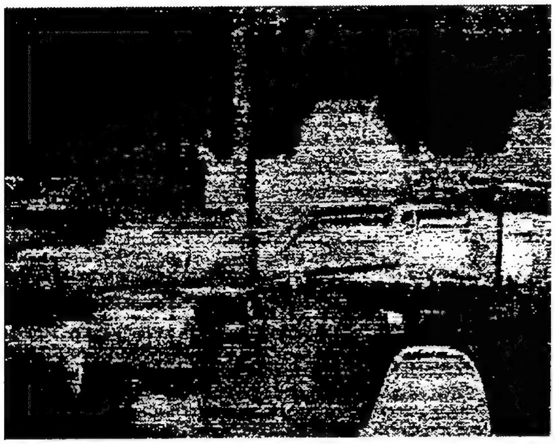
(((((b51)+((b29)+(((b77)-((b87)-(b27)))/(b80))))+(b95))+((((((b14)+(b21))+((((((((b76)*(b48))*(b21))
(((b6)+(b111))+(b98)))((b103)+(b95)))*(((((((b94)-(b24))/((b33)/(b32)))*((b29)-(b35)))*((b79)/(b95)))
(((b2)+(b18))(((b76)*(b48))/(b70))))+((b10)-(b4))))+((((((b60)*(b56))+(b111))+(b96))-(((b95)/(b10))
-((b48)/(b70))))+((((b2)-(b29))+(b111))+((b54)*(b91)))))*(b55)))+((((b44)+(b108))-(((b10)-(b91))
-((((b103)+(b95))+((b112)+(b17)))-((b10)-((b48)/(b70))))))-((((b98)+(b41))/((((((b77)-((b87)-(b27)))
+((b103)+(b95)))-(((b95)/(b10))-((b39)-(b14))))-(((b10)-(b4))/((b33)/(b32))))/(b70)))-((b9)-(b41)))))
-((((b87)-(b27))/(b10))-((b48)/(b70))))+((b25)*((b44)+(b108)))))-((b10)-((b48)/(b70))))

Barrel-Dark equation 9

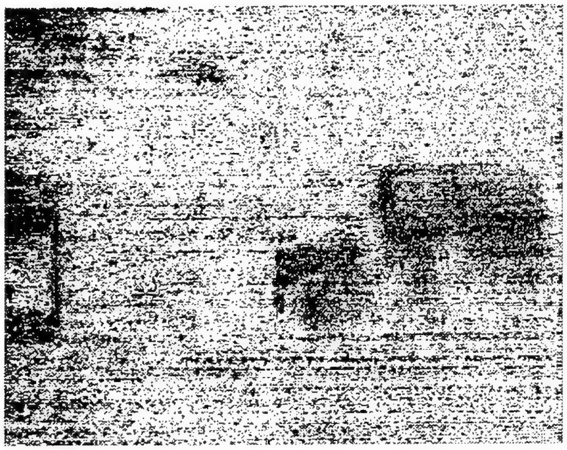


Barrel image

Fence image



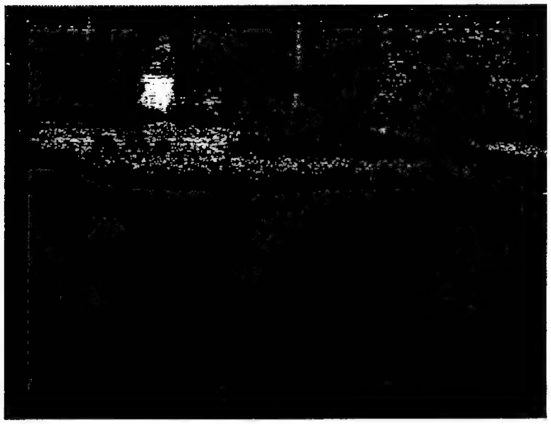
Trucks image



Van image

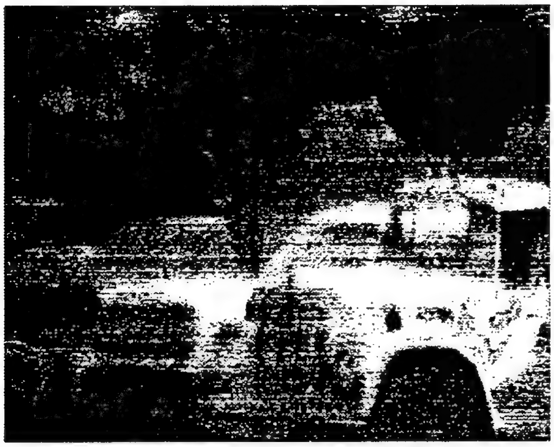
```
(((b81)-(((b71)-((((((b7)/(b4))-((b73)-(b9)))+(((b108)-(b69))+((b11)+((b5)/(b100)))))+(((b4)+(b12))
+(((b8)/(b15))*((b13)/(b21)))))-(((((b51)*((((b77)+(b1))/(((b35)*(b53))-(((b21)-(b93))-((b73)-(b107)))))
+((b73)-(b9))))*((b6)*(b21)))/((((b74)/((b111)-(b72)))/(((b7)/(b4))-(((b6)*(b21))+(b78))))+(((b7)/(b4))
-((b20)+((((b79)-(b5))+((b73)-(b107)))/(((b25)+(b98))+(((b93)+(b47))*((b44)+(b23))))))))-(((b36)
+(b99))/((((b112)*(b13))/((b112)-(b55)))*((((((b8)/(b15))*((b13)/(b21)))+((b79)-(b5)))/(((b73)-(b9))
-((b25)+(b98))))/((b79)*(b22))))))))+(b97)))-(((b29)+(b105))*((((b111)-(b34))+((b33)+((b44)
+(((((b51)*((((b77)+(b1))/(((b35)*(b53))-((b7)/(b4))))+((b112)-(b55))))*((b6)*(b21)))/((((b74)/((b111)
-(b72)))/((b6)*(b21)))+(((b7)/(b4))-((b20)+((((b79)-(b5))+(((b112)*(b13))/((b112)-(b55))))
/(((b25)+(b98))+(((b7)/(b4))*((b44)+(b23)))))))))*(b10)))))-((b58)/((((b77)-(b92))*((b37)-(b71)))
-(((b70)*(b70))+((b21)-(b93))))))))
```

Barrel-Bright equation 13

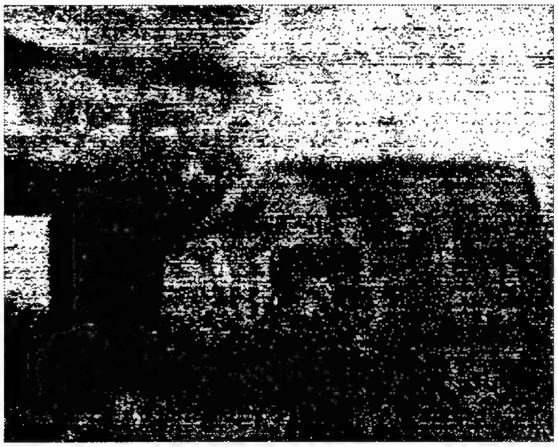


Barrel image

Fence image



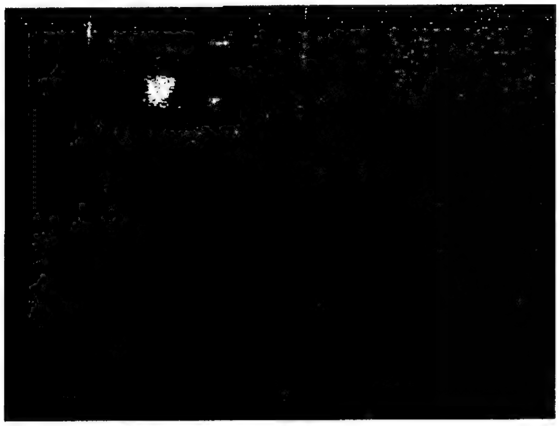
Trucks image



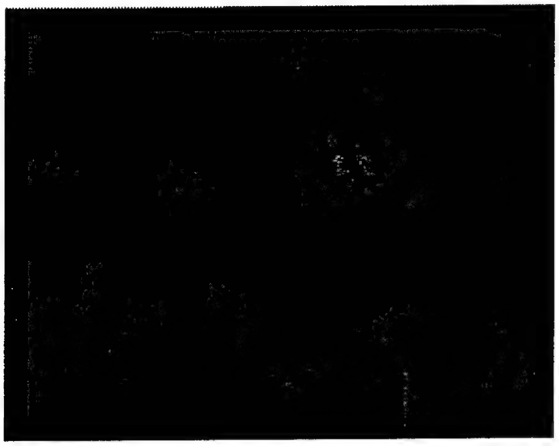
Van image

(((b81)-(((b71)-((((((b7)/(b4))-((b73)-(b9)))+(((b108)-(b69))+((b11)+((b5)/(b100)))))+(((b4)+(b12))
+(((b8)/(b15))*((b13)/(b21)))))-(((((b51)*((((b77)+(b1))/(((b35)*(b53))-(((b21)-(b93))-((b73)-(b107)))))
+((b73)-(b9))))*((b6)*(b21)))/((((b74)/((b111)-(b72)))/(((b7)/(b4))-(((b6)*(b21))+(b78))))+(((b7)/(b4))
-((b20)+((((b79)-(b5))+((b73)-(b107)))/(((b25)+(b98))+(((b93)+(b47))*((b44)+(b23)))))))))
-(((b36)+(b99))/((((b112)*(b13))/((b112)-(b55)))*((((((b8)/(b15))*((b13)/(b21)))+((b79)-(b5)))/(((b73)
-(b9))-((b25)+(b98))))/((b79)*(b22))))))))+(b97)))-(((b29)+(b105))*((((b111)-(b34))+((b33)+((b44)
+(((((b51)*((((b77)+(b1))/(((b35)*(b53))-((b7)/(b4))))+((b112)-(b55))))*((b6)*(b21)))/((((b74)/((b111)
-(b72)))/((b6)*(b21)))+(((b7)/(b4))-((b20)+((((b79)-(b5))+(((b112)*(b13))/((b112)-(b55))))/(((b25)
+(b98))+(((b7)/(b4))*((b44)+(b23))))))))))*(b10)))))-((b58)/((((b77)-(b92))*((b37)-(b71)))-(((b70)
*(b70))+((b21)-(b93))))))))

Barrel-Bright equation 14

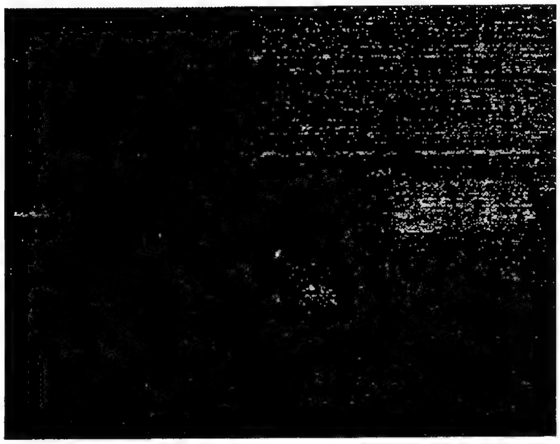


Barrel image



Fence image

Trucks image



Van image

```
(((b81)-(((b71)-((((((b7)/(b4))-((b73)-(b9)))+(((b108)-(b69))+((b11)+((b5)/(b100)))))+(((b4)+(b12))
+(((b8)/(b15))*((b13)/(b21)))))-(((((b51)*((((b77)+(b1))/(((b35)*(b53))-(((b21)-(b93))-((b73)-(b107)))))
+((b73)-(b9)))))*((b6)*(b21)))/((((b74)/((b111)-(b72)))/(((b7)/(b4))-(((b6)*(b21))+(b78))))+(((b7)/(b4))
-((b20)+((((b79)-(b5))+((b73)-(b107)))/(((b25)+(b98))+(((b93)+(b47))*((b44)+(b23)))))))))-(((b36)
+(b99))/((((b112)*(b13))/((b112)-(b55)))*((((((b8)/(b15))*((b13)/(b21)))+((b79)-(b5)))/(((b73)-(b9))
-((b25)+(b98))))/((b79)*(b22))))))))+(b97)))-(((b29)+(b105))*((((b111)-(b34))+((b33)+((b44)
+(((((b51)*((((b77)+(b1))/(((b35)*(b53))-((b7)/(b4))))+((b112)-(b55))))*((b6)*(b21)))/((((b74)/((b111)
-(b72)))/((b6)*(b21)))+(((b7)/(b4))-((b20)+((((b79)-(b5))+(((b112)*(b13))/((b112)-(b55))))/(((b25)+(b98))
+(((b7)/(b4))*((b44)+(b23)))))))))*(b10)))))-((b58)/((((b77)-(b92))*((b37)-(b71)))-(((b70)*(b70))+((b21)
-(b93))))))))
```

Distribution

Admnstr
Defns Techl Info Ctr
ATTN DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

DARPA
ATTN S Welby
3701 N Fairfax Dr
Arlington VA 22203-1714

Ofc of the Secy of Defns
ATTN ODDRE (R&AT)
The Pentagon
Washington DC 20301-3080

AMCOM MRDEC
ATTN AMSMI-RD W C McCorkle
Redstone Arsenal AL 35898-5240

AVCOM
ATTN AMSAM-RD-WS-PL W Davenport
Redstone Arsenal AL 35898

US Army TRADOC
Battle Lab Integration & Techl Dirctrt
ATTN ATCD-B
FT Monroe VA 23651-5850

US Military Acdmy
Mathematical Sci Ctr of Excellence
ATTN MADN-MATH MAJ M Johnson
Thayer Hall
West Point NY 10996-1786

Dir for MANPRINT
Ofc of the Deputy Chief of Staff for Prsnnl
ATTN J Hiller
The Pentagon Rm 2C733
Washington DC 20301-0300

SMC/CZA
2435 Vela Way Ste 1613
El Segundo CA 90245-5500

TECOM
ATTN AMSTE-CL
Aberdeen Proving Ground MD 21005-5057

US Army ARDEC
ATTN AMSTA-AR-TD
Bldg 1
Picatinny Arsenal NJ 07806-5000

US Army CECOM RDEC NVESD
ATTN NVESD-RP-FS C Simi
ATTN T Colandini
10221 Burbech Rd Ste 430
FT Belvoir VA 22060-5806

US Army Info Sys Engrg Cmnd
ATTN AMSEL-IE-TD F Jenia
FT Huachuca AZ 85613-5300

US Army Natick RDEC Acting Techl Dir
ATTN SBCN-T P Brandler
Natick MA 01760-5002

US Army Simulation Train & Instrmntn Cmnd
ATTN AMSTI-CG M Macedonia
ATTN J Stahl
12350 Research Parkway
Orlando FL 32826-3726

US Army Tank-Automtv Cmnd RDEC
ATTN AMSTA-TR J Chapin
Warren MI 48397-5000

US Army Topograpphic Engrg Ctr
ATTN CEERD-TR-S S Barr
ATTN R Rand
7701 Telegraph Rd
Alexandria VA 22315

Nav Surfc Warfare Ctr
ATTN Code B07 J Pennella
17320 Dahlgren Rd Bldg 1470 Rm 1101
Dahlgren VA 22448-5100

CMTCO
ATTN MAJ A Suzuki
1030 S Highway A1A
Patrick AFB FL 23925-3002

Distribution (cont'd)

SMDC
ATTN SMDC-TC-TD-YF A Aberle
Redstone Arsenal AL 35898

Univ of Maryland
ATTN G Syrmos
Martin Hall, Rm 3136
College Park MD 20742-3011

Univ of Michigan
ATTN J Daida
2455 Hayward Ave 2517D Space Rsrch Bldg
Ann Arbor MI 48109-2143

Aerospace Corp
ATTN M Hamilton
15049 Conf Ctr Dr Ste 600
Chantilly VA 20151-3824

ERIM
ATTN E Crist
ATTN J Ackenhusen
1975 Green Rd
Ann Arbor MI 48105

Hicks & Assoc Inc
ATTN G Singley III
1710 Goodrich Dr Ste 1300
McLean VA 22102

Sensor Tech & Sys Appl Group
MIT Lincoln Lab
ATTN J P Kerekes
244 Wood St
Lexington MA 02420-9185

NIMA ATTR
ATTN J Hammack
4600 Sangemore Rd Mail Stop D65
Bethesda MD 20816-5003

Palisades Inst for Rsrch Svc Inc
ATTN E Carr
1745 Jefferson Davis Hwy Ste 500
Arlington VA 22202-3402

Director
US Army Rsrch Lab
ATTN AMSRL-RO-D JCI Chang
ATTN AMSRL-RO-EN B Mann
PO Box 12211
Research Triangle Park NC 27709

US Army Rsrch Lab
ATTN AMSRL-D D R Smith
ATTN AMSRL-DD J M Miller
ATTN AMSRL-CI-EP A Wetmore
ATTN AMSRL-CI-IS-R Mail & Records Mgmt
ATTN AMSRL-CI-IS-T Techl Pub (2 copies)
ATTN AMSRL-CI-OK-TL Techl Lib (2 copies)
ATTN AMSRL-SE-EE N Gupta
ATTN AMSRL-SE-R B Wallace
ATTN AMSRL-SE-SE H Kwon
ATTN AMSRL-SE-SE N Nasrabadi
ATTN AMSRL-SE-SE P Rauss (15 copies)
ATTN AMSRL-SE-SE S Der
ATTN AMSRL-SE-SE T Kipp
ATTM AMSRL-SE-SE J Hackett
ATTN AMSRL-SL-BE P Tanenbaum
Adelphi MD 20783-1197

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** November 2001	**3. REPORT TYPE AND DATES COVERED** Final, Oct 1999 to Sept 2000

4. TITLE AND SUBTITLE Hyperspectral and Spectro-Polarimetric Pixel-Level Classification Using Genetic Programming

5. FUNDING NUMBERS
DA PR: AH16
PE: 62120A

6. AUTHOR(S) Patrick Rauss

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL-SE-SE email: prauss@arl.army.mil
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-907

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
ARL PR: 1NE4M3
AMS code: 622120.H1611

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The objective force will be relying heavily on their sensors to be a combat multiplier to help improve the force's effectiveness and survivability, particularly for reconnaissance, surveillance, and target acquisition missions. Currently, fielded passive sensor systems are generally ineffective against camouflage, concealment, and deception. Their performance is also sensitive to environmental conditions. To meet future needs, several new sensor systems are being developed and evaluated. Two of these new sensors are passive systems that collect additional, measurable characteristics of light: hyperspectral (HS) systems and spectro-polarimetric (SP) systems.

To fully take advantage of the information that these systems collect requires new algorithms and techniques. This report discusses why new techniques are necessary and details the development of a computer-assisted design system for the discovery of classification algorithms via a small number of sample target and background signatures. The technique is called genetic programming(GP). GP is an adaptive learning technique that automatically generates a computer program (in this work, a mathematical equation) to solve the problem it is given.

14. SUBJECT TERMS Hyperspectral, genetic programming, spectral classification, material	**15. NUMBER OF PAGES** 79
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

13. ABSTRACT (cont'd)

This report documents work conducted primarily between September 1999 and August 2000, while the author was on a rotation at the University of Michigan under the Federated Laboratories Consortium program. The report demonstrates that GP could be a useful technique for processing HS and SP data. The experiments reported here show that by using even the simplest of operators (addition, subtraction, multiplication and division) the GP process can develop interesting and potentially useful solution equations. The results shown here are encouraging. However, many questions remain to be answered.